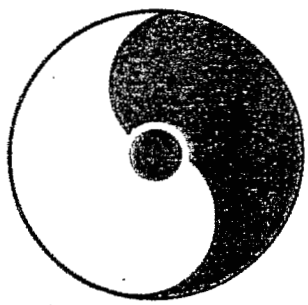
BNL-71147-2003
Formal Report

Proceedings of RIKEN BNL Research Center Workshop Volume 50

High Performance Computing with QCDOC and BlueGene

February 28, 2003



Organizers:

N. Christ (Columbia Univ), J. Davenport (BNL), Y. Deng (BNL/Stony Brook Univ), A. Gara (IBM), J. Glimm (BNL/Stony Brook Univ), R. Mawhinney (Columbia Univ), E. McFadden (BNL), A. Peskin (BNL), W. Pulleyblank (IBM)

RIKEN BNL Research Center

Building 510A, Brookhaven National Laboratory, Upton, NY 11973-5000, USA

DISCLAIMER

Available electronically at-

`http://www.doe.gov/bridge`

Available to U.S. Department of Energy and its contractors in paper from-

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831
(423) 576-8401

Available to the public from-

U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Road
Springfield, VA 22131
(703) 487-4650

Printed on recycled paper

Preface to the Series

The RIKEN BNL Research Center (RBRC) was established in April 1997 at Brookhaven National Laboratory. It is funded by the "Rikagaku Kenkyusho" (RIKEN, The Institute of Physical and Chemical Research) of Japan. The Center is dedicated to the study of strong interactions, including spin physics, lattice QCD, and RHIC physics through the nurturing of a new generation of young physicists.

During the first year, the Center had only a Theory Group. In the second year, an Experimental Group was also established at the Center. At present, there are seven Fellows and seven Research Associates in these two groups. During the third year, we started a new Tenure Track Strong Interaction Theory RHIC Physics Fellow Program, with six positions in the first academic year, 1999-2000. This program had increased to include ten theorists and one experimentalist in academic year, 2001-2002. With recent graduations, the program presently has eight theorists and two experimentalists. Beginning last year a new RIKEN Spin Program (RSP) category was implemented at RBRC, presently comprising four RSP Researchers and five RSP Research Associates. In addition, RBRC has four RBRC Young Researchers.

The Center also has an active workshop program on strong interaction physics with each workshop focused on a specific physics problem. Each workshop speaker is encouraged to select a few of the most important transparencies from his or her presentation, accompanied by a page of explanation. This material is collected at the end of the workshop by the organizer to form proceedings, which can therefore be available within a short time. To date there are forty-nine proceeding volumes available.

The construction of a 0.6 teraflops parallel processor, dedicated to lattice QCD, begun at the Center on February 19, 1998, was completed on August 28, 1998. A 10 teraflops QCDOC computer in under development and expected to be completed in JFY 2003.

T. D. Lee
November 22, 2002

*Work performed under the auspices of U.S.D.O.E. Contract No. DE-AC02-98CH10886.

CONTENTS

High Performance Computing With QCDOC and BlueGene

Introduction

Staff of Brookhaven National Laboratory, Columbia University, IBM and the RIKEN BNL Research Center organized a one-day workshop held on February 28, 2003 at Brookhaven to promote the following goals:

1) To explore areas other than QCD applications where the QCDOC and BlueGene/L machines can be applied to good advantage,
2) To identify areas where collaboration among the sponsoring institutions can be fruitful, and
3) To expose scientists to the emerging software architecture.

This workshop grew out of an informal visit last fall by BNL staff to the IBM Thomas J. Watson Research Center that resulted in a continuing dialog among participants on issues common to these two related supercomputers. The workshop was divided into three sessions, addressing the hardware and software status of each system, prospective applications, and future directions.

The first session was divided into four presentations, updating the hardware and software developments for the QCDOC and BlueGene/L systems, respectively. Norman Christ presented the QCDOC overview and status. Peter Boyle, Dave Stampf and Robert Bennett described the user software environment for that machine, including user extensions such as MPI. Dong Chen and Jose Moreira gave overviews of the BlueGene/L hardware and software. The machines are similar in their torus interconnect network, their underlying chip technology and in the topical nature of their target applications. They differ primarily in that BlueGene is designed to have a wider range of applications, while QCDOC is expected to be ready first. Hence, the developments of each are of considerable interest to the constituencies of both systems.

The second session highlighted some of the applications under consideration, particularly those other than the intended applications. Jim Glimm described the work that has been done analyzing the suitability of QCDOC for molecular dynamics modeling. Doug Swesty described QCDOC's prospects in astrophysics. Bob Germain described the suite of applications that have been studied for BlueGene/L. David Keyes discussed the attributes of supercomputer architectures required for solving parallel implicit PDEs, and Mark Tuckerman described optimal schemes for Car-Parrinello based *ab initio* molecular dynamics studies.

The final session looked to the future. Al Gara highlighted the similarities and differences between the two systems. Bill Gropp helped place these systems in the context of next generation supercomputer developments, including commodity, vector and topical architectures. Jim Davenport hosted a closing discussion of future work, including

identification of independent activities that might be strengthened through collaboration, possible avenues for this community to continue to be involved with each other, and promoting means of encouraging more widespread access to machine simulation studies. There appeared to be a consensus that this workshop had been valuable, and that there should be some formal follow-up, perhaps at a BlueGene workshop at Livermore this fall already in the planning stage.

Organizing Committee:

Norman Christ - Columbia
Jim Davenport – BNL
Yuefan Deng – BNL/Stony Brook
Al Gara - IBM
Jim Glimm – BNL/Stony Brook
Bob Mawhinney - Columbia
Ed McFadden - BNL
Arnie Peskin – BNL
Bill Pulleyblank – IBM

Only Difference Equations

T. D. Lee

Columbia University and RBRC

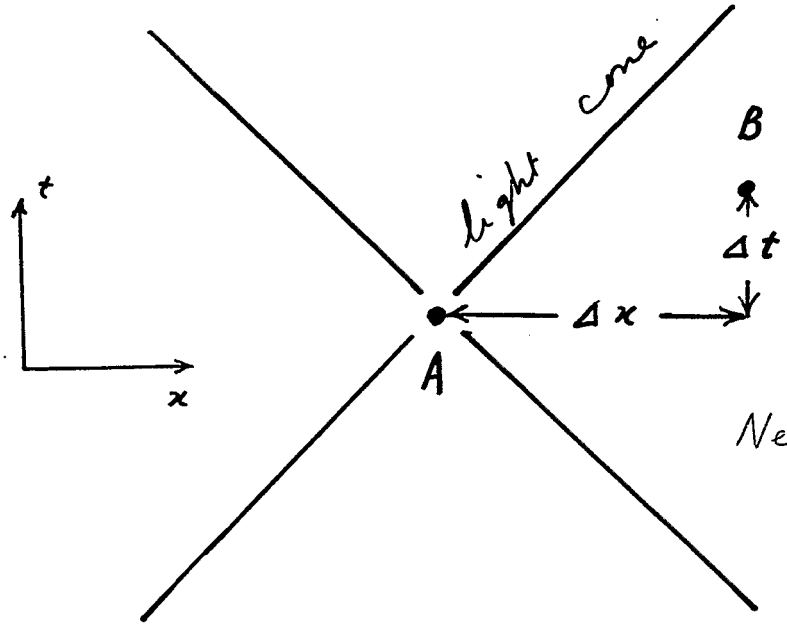
Presented at the QCDOC/Blue Genes Workshop

February 28, 2003

Fundamental Physics should be based on Difference Equations (not differential equations)

- Local field theory is inadequate
- **Both difference and differential equations can have the same continuous groups of sym. (including translations and rotations)**
- Difference equations have chaos and fractal type solutions, not possessed by differential equations
- **Differential equations are only approximations to difference equations**
- Physics should be described by difference equations, not differential equations.

two local measurements at A and B



$\hbar = c = 1$

Newton's const. $G = l_p^2$

$\Delta t < \Delta x <$ Planck length $l_p \sim 10^{-33}$ cm

local field theory states that A and B are independent

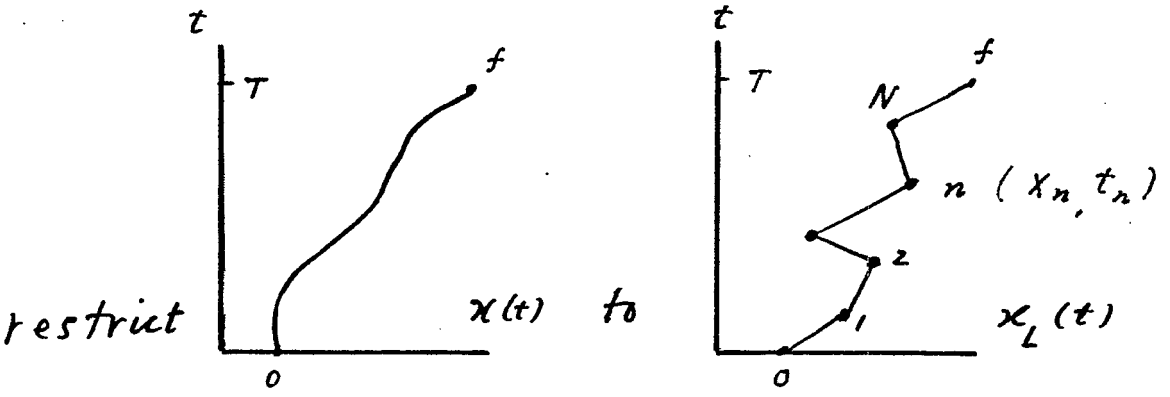
But $\Delta E \sim \frac{1}{\Delta t} > \frac{1}{l_p}$

black hole horizon rad. $\sim G\Delta E > G\frac{1}{l_p}$

$\therefore$ " $> l_p > \Delta x$!

How can A & B be independent ?

example : Classical Mech.



action $A(x(t)) = \int \left[\frac{1}{2} \dot{x}^2 - V(x) \right] dt$

becomes $A_L = A(x_L(t))$, $\boxed{\frac{N}{T} \text{ fixed}}$

$\therefore$ time translational inv.

$$A_L = \sum_n \left\{ \frac{1}{2} \frac{(x_n - x_{n-1})^2}{t_n - t_{n-1}} - \frac{1}{2}\left[V(x_n) + V(x_{n-1})\right] \cdot (t_n - t_{n-1}) \right\}$$

$\delta A_L = 0$ gives exact energy cons.

& momentum cons.

$$\frac{\partial A_L}{\partial x_n} = 0 : \quad v_n - v_{n+1} = -\frac{1}{2}(t_{n-1} - t_{n+1}) \frac{\partial V(x_n)}{\partial x_n} .$$

where $v_n = (x_n - x_{n-1}) / (t_n - t_{n-1})$

$$\frac{\partial A_L}{\partial t_n} = 0 : \quad E_n = \frac{1}{2} v_n^2 + \frac{1}{2}\left[V(x_n) + V(x_{n-1}) \right]$$

$$= E_{n+1} .$$

General Relativity

$$A_S = \int_S \sqrt{|g|}\, R\, d^D x$$

S = arbitrary D-dim smooth manifold

Discrete Gravity

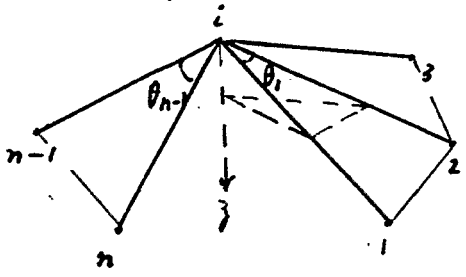
Restrict S to L = D-dim continuous piecewise flat surface of D-simplices

$$A_L = \int_L \sqrt{|g|}\, R\, d^D x$$

Both A_L and A_S are inv. under $x_\mu \to x'_\mu$

---

example $D = 2$ Embed L in R_3



$R_3 = (x, y, z)$ space

Δ = 2 - simplex

$$A_L = \sum_i 2\epsilon_i$$

$\epsilon_i = 2\pi - (\theta_1 + \theta_2 + \cdots + \theta_n)$

= deficit angle

Embed L in R_N, $\min N =$

D =	2	3	4
	3	7	19

Thm For any L of D-dim

$$A_L = \int_L \sqrt{|g|}\, R\, d^D x = 2 \sum_s \epsilon_s$$

(Regge Calculus)

s = vol of $D-2$ simplex

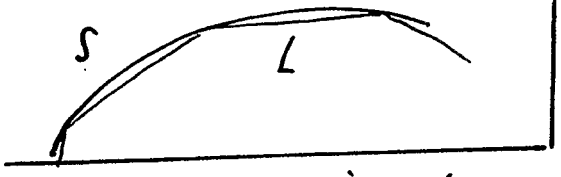
ϵ_s = deficit angle around s

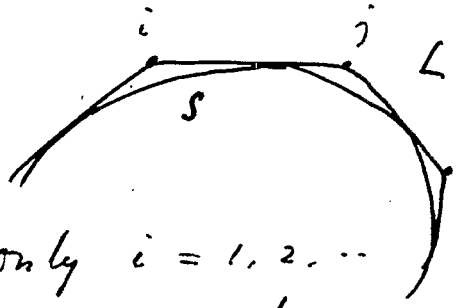
$D = 2$ $\quad A_L = 2 \sum_i \epsilon_i$ (Gauss-Bonnet)

3 $\quad = 2 \sum_l l\, \epsilon_l$ (TDL and R. Friedberg)

4 $\quad = 2 \sum_\Delta \Delta\, \epsilon_\Delta$

Regge's idea	Discrete Gravity
Fix S	Fix L
vary $L \to S$	vary $S \to L$





Discrete gravity has more sym!

$\therefore$ only $i = 1, 2, \ldots$ are real

RHIC

QCDOC
System Overview and Project Status

Norman H. Christ
Columbia Univeristy
New York, NY 10027

The QCDOC (Quantum Chromodynamics of a Chip) computer architecture is intended to provide a cost-effective computing platform for very large scale lattice QCD calculations. By utilizing a 6-dimensional mesh interconnection network and simple computing nodes constructed from a single chip and standard memory module we are able to provide a machine capable of scaling efficiently to tens of thousands of nodes and delivering more than 10 Teraflops of sustained performance at a cost per sustained performance of less than \$1/Mflops.

An industry standard, PowerPC RISC processor, 4 Mbytes of memory, two Ethernet ports and 24, 500 Mbit/s serial links are included in the chip which forms the basis of the computational node. The resulting compact design with very few components permits very large-scale, low-power systems to be constructed. Extensive built-in error checking and correcting for the serial communications and both the on- and off-chip memory increases the reliability of the machine.

Considerable attention is paid to the bandwidth and latency of both the memory system and the serial communications. Sufficient independent control circuitry is provide to permit a large overlap between the nearest-neighbor serial communication and on-node computations. As a result very good performance is achieved for QCD even when a fixed size system is studied using an increasingly large machine. The table below shows performance at approximately 50% of peak for calculations on a fixed $32^3 \times 64$ lattice size as the machine size is increased from 4K to 32K nodes and the number of lattice points per node falls to a very small 64.

Nodes	$M^\dagger M$+linalg	Global Sum	Sust. Tflops
4096	2620μs	10 μs	2.15
8192	1310μs	11.5μs	4.2
16384	680 μs	13 μs	8.1
32768	340 μs	15 μs	15.6

We are now in the final stages of the design of this ASIC with tape out expected in March and prototype chips in May. We plan substantial development machines (7 Teraflops peak in aggregate) in Fall 2003. Large scale, 10 Teraflops machines will be available at the RBRC here at Brookhaven and at the University of Edinburgh in 2004. Finally a 20 Teraflops (peak) DOE-funded machine is planned for Brookhaven also in 2004. These machines have been designed to achieve high efficiency for lattice QCD calculations. However, we hope that they may also be effective for other targeted applications which are able to exploit the nearest-neighbor grid-based communications network.

STRATEGY

- QCD Requirements:
  - Space-time homogeneity supports easy parallelization and a mesh network.
  - Scaling implies small volumes/node:

$$\begin{aligned} \text{Work} &\propto N_{\text{sites}}^3 \\ \text{Power} &\propto N_{\text{processors}} \end{aligned}$$

Fixed execution speed requires:

$$\begin{aligned} \frac{N_{\text{sites}}}{N_{\text{processors}}} &\propto N_{\text{sites}}/N_{\text{sites}}^3 \\ &\propto 1/N_{\text{sites}}^2 \propto 1/L^8 \end{aligned}$$

  - Good scaling allows optimization of:
    - processor price/performance
    - power
    - packaging
  - Good scaling requires:
    - high bandwidth
    - low latency
    - small packet size

COLLABORATION

Columbia (DOE): Norman Christ
Saul Cohen
Calin Cristian
Zhihua Dong
Valeriya Gadiyak
Changhoan Kim
Ludmilia Levkova
Xiaodong Liao
HueyWen Lin
Guofeng Liu
Robert Mawhinney
Azusa Yamaguchi

BNL (SciDAC): Robert Bennett
Tameka Carter
Chulwoo Jung
Konstantin Petrov
David Stampf

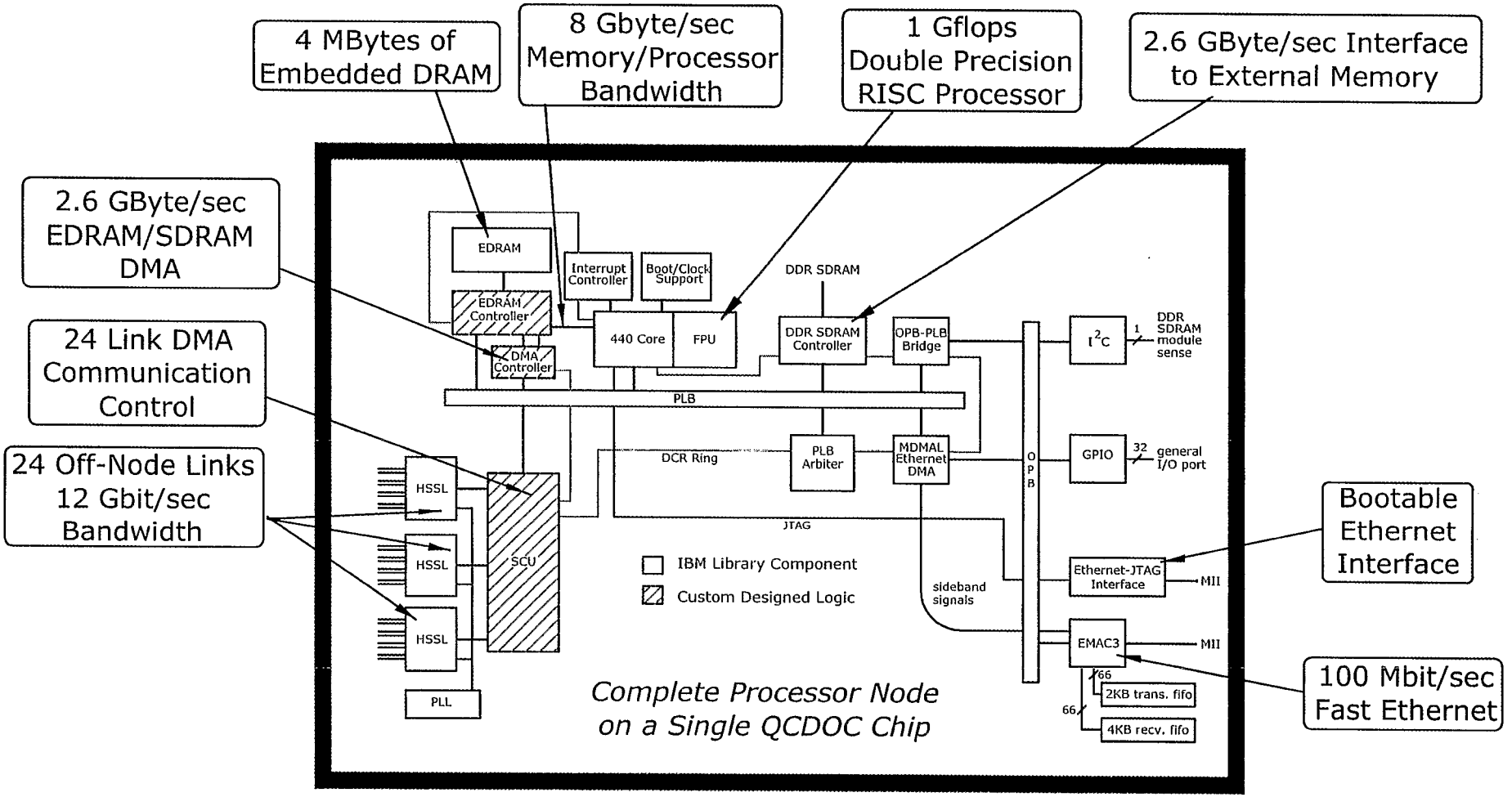
UKQCD (PPARC): Peter Boyle
Balint Joo

RBRC (RIKEN): Shigemi Ohta (KEK)
Tilo Wettig (Yale)

IBM: Dong Chen
Alan Gara
Design groups:
Yorktown Heights, NY;
Rochester, MN; Raleigh, NC

DESIGN

- IBM-fabricated, single-chip node.
  [50 million transistors, 3-4 Watt, 1.3cm×1.3cm die]
- PowerPC 32-bit processor
  - 1 Gflops, 64-bit IEEE FPU.
  - Memory management.
  - GNU and XLC compilers.
- 4 Mbyte on-chip memory and up to 2.0 Gbyte/node on DIMM card.
- 6-dim communications network:
  - Efficient for small packet sizes, $\approx$ 200ns latency.
  - Global sum/broadcast functionality.
  - Minimal processor overhead.
  - Lower dimensional machine partitions.
- 100 Mbit/sec, Fast Ethernet
  - JTAG/Ethernet boot hardware.
  - Host-node OS communication.
  - Disk I/O.
  - RISCWatch debugger.
- ≈ 5 Watt, 15 in^3 per node.


4 MBytes of Embedded DRAM
8 Gbyte/sec Memory/Processor Bandwidth
1 Gflops Double Precision RISC Processor
2.6 GByte/sec Interface to External Memory
2.6 GByte/sec EDRAM/SDRAM DMA
24 Link DMA Communication Control
24 Off-Node Links 12 Gbit/sec Bandwidth
Bootable Ethernet Interface
100 Mbit/sec Fast Ethernet
EDRAM
EDRAM Controller
DMA Controller
Interrupt Controller
Boot/Clock Support
440 Core
FPU
DDR SDRAM
DDR SDRAM Controller
OPB-PLB Bridge
PLB
DCR Ring
PLB Arbiter
MDMAL Ethernet DMA
OPB
I2C
1
DDR SDRAM module sense
GPIO
32
general I/O port
JTAG
sideband signals
Ethernet-JTAG Interface
MII
EMAC3
MII
66
2KB trans. fifo
66
4KB recv. fifo
HSSL
HSSL
HSSL
SCU
PLL
IBM Library Component
Custom Designed Logic
Complete Processor Node on a Single QCDOC Chip

COMMUNICATIONS SPECIFICS

- 64-bit packets with 8-bit headers.
- Each packet acknowledged but "3-in-the-air" allowed.
- Send and receive in each of the 12 directions driven by block-strided DMA — 24 channels.
- 16-deep, chained DMA instruction memory for each channel.
- Single PowerPC store launches DMA-driven communications on up to 24 channels.
- Polled or interrupt-driven completion.
- Independent tranmission/reception of 64-bit, memory-mapped supervisor packets.
- Store-and-forward capability:
  - Receive data stream from one direction.
  - DMA-driven storage of incoming data stream.
  - Multi-direction broadcast of incoming data stream.
  - Data stream broadcast preceded by up-to 128-bits of local data.
  - Two identical store-and-forward engines allow only L/2 operations.

QCDOC Operating System

Peter Boyle
University of Edinburgh
Columbia University

- Overview
- Node-kernels
- Qdaemon
- Software Partitioning

QCDOC Operating System

Peter Boyle
University of Edinburgh
Columbia University

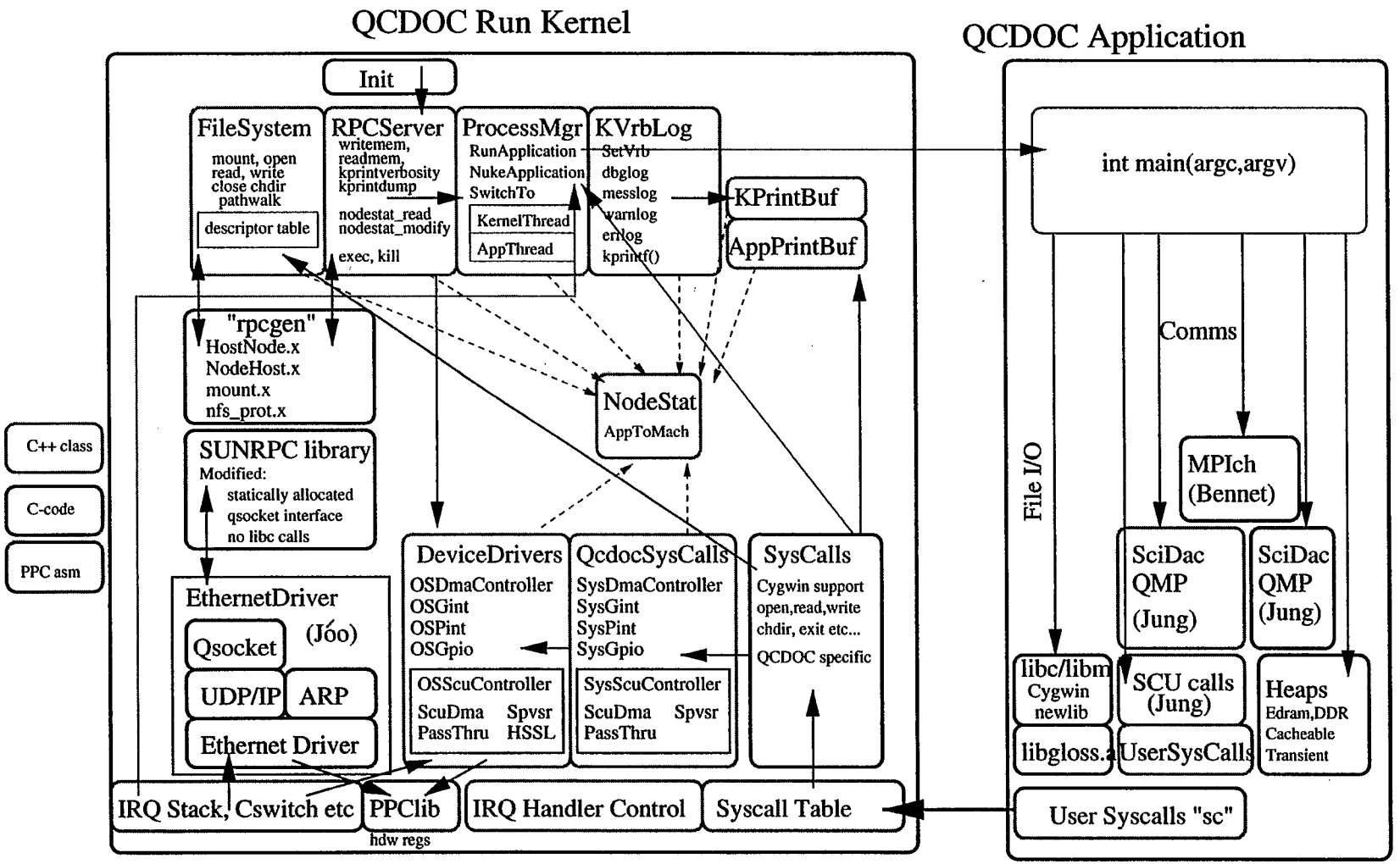
We discuss the software strategy and operating system for QCDOC. In particular we focus on the Qdaemon management software on the front end computer, the node kernels, and the partitioning scheme.

Operating System

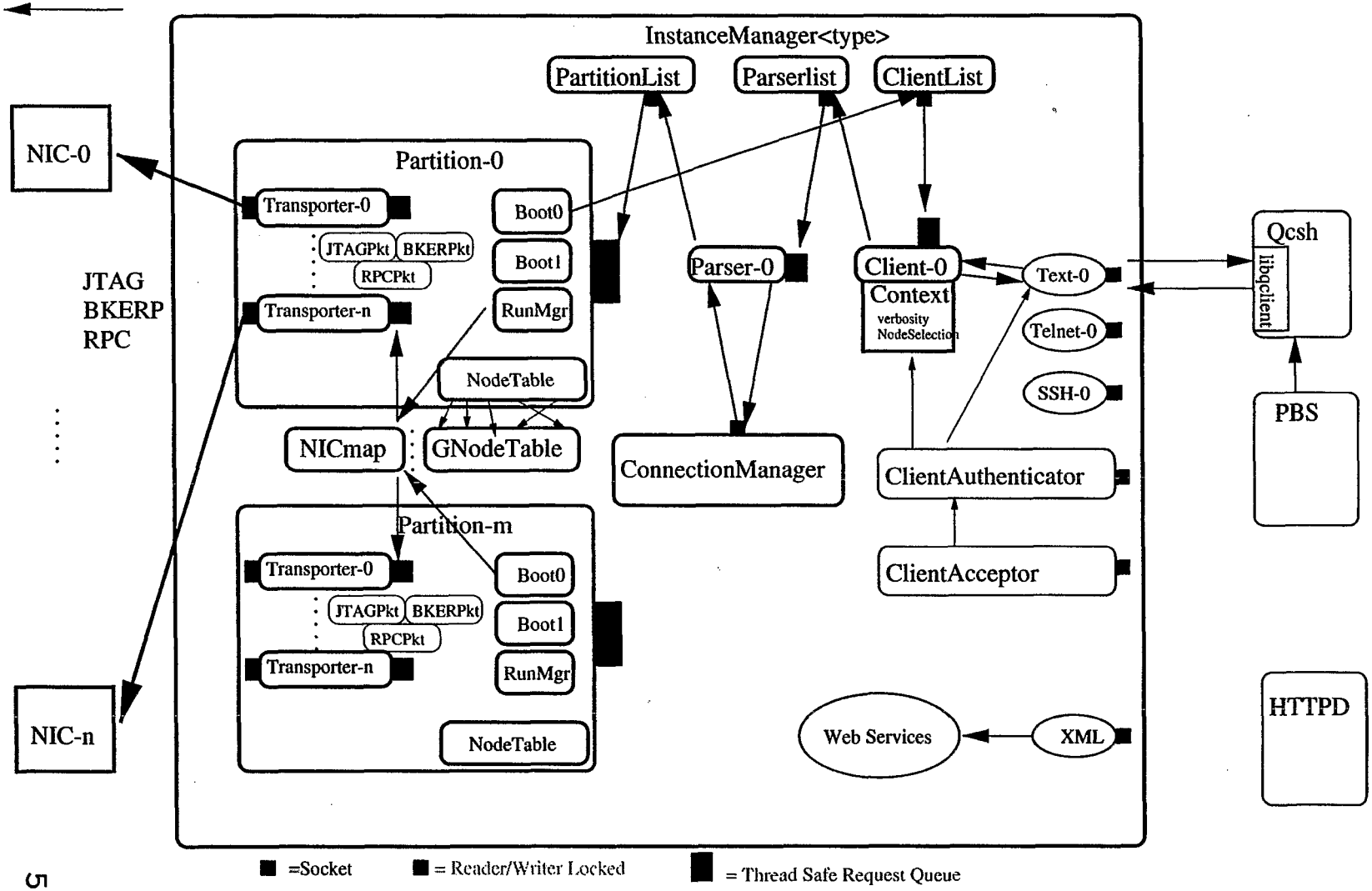
- Boot machine
- Diagnose hardware errors
- Run one program on each CPU
- Service Comms and I/O requests

Robust, Simple, Efficient
Debug hardware with software!

- No-nonsense front end.
- Controlled boot sequence + lean kernel
- Run one process and run it well.
  VERY tightly coupled - no scheduling.
- Memory protection but not translation
  Zero copy DMA + Never miss TLB.


QCDOC Run Kernel
QCDOC Application
Init
FileSystem
mount, open
read, write
close chdir
pathwalk
descriptor table
RPCServer
writemem,
readmem,
kprintverbosity
kprintdump
nodestat_read
nodestat_modify
exec, kill
ProcessMgr
RunApplication
NukeApplication
SwitchTo
KernelThread
AppThread
KVrbLog
SetVrb
dbglog
messlog
warnlog
errlog
kprintf()
KPrintBuf
AppPrintBuf
int main(argc,argv)
"rpcgen"
HostNode.x
NodeHost.x
mount.x
nfs_prot.x
NodeStat
AppToMach
Comms
File I/O
MPIch
(Bennet)
SUNRPC library
Modified:
statically allocated
qsocket interface
no libc calls
C++ class
C-code
PPC asm
DeviceDrivers
OSDmaController
OSGint
OSPint
OSGpio
OSScuController
ScuDma Spvsr
PassThru HSSL
QcdocSysCalls
SysDmaController
SysGint
SysPint
SysGpio
SysScuController
ScuDma Spvsr
PassThru
SysCalls
Cygwin support
open,read,write
chdir, exit etc...
QCDOC specific
SciDac
QMP
(Jung)
SciDac
QMP
(Jung)
EthernetDriver
Qsocket
(Jóo)
UDP/IP
ARP
Ethernet Driver
libc/libm
Cygwin
newlib
libgloss.a
SCU calls
(Jung)
UserSysCalls
Heaps
Edram,DDR
Cacheable
Transient
IRQ Stack, Cswitch etc
PPClib
hdw regs
IRQ Handler Control
Syscall Table
User Syscalls "sc"

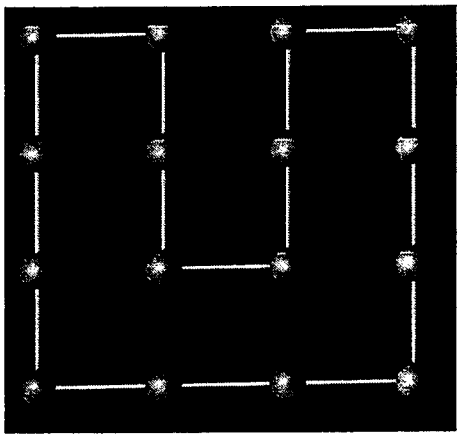
Qdaemon Architecture



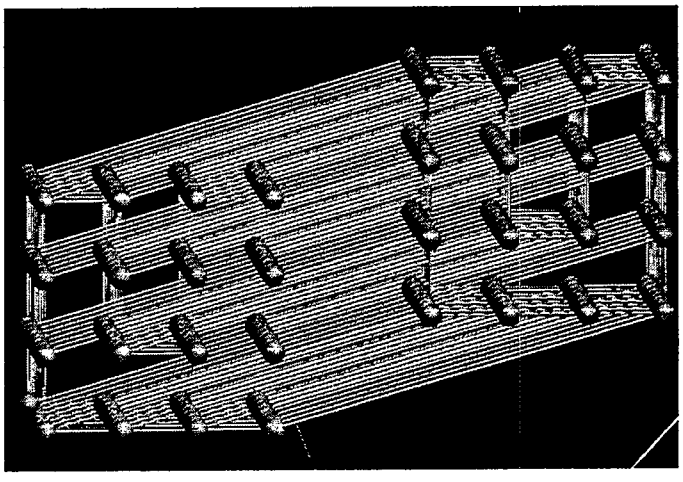
Dimension Folding

Simple case:
Fold two machine axes m_1, m_2 lengths l_1, l_2
Forms one application axis length $l_1 \times l_2$
Requirement: one of l_1 or l_2 is even.

$l_1 \cdot l_2 \cdot l_3 \cdot l_4 \cdot l_5 \cdot l_6 \rightarrow (l_1 \times l_2) \cdot l_3 \cdot l_4 \cdot l_5 \cdot l_6 \cdot 1$



$4 \cdot 4 \longrightarrow 16$



$4 \cdot 4 \cdot 8 \cdot 2 \longrightarrow 16 \cdot 8 \cdot 2$

Remapped partition is a properly connected grid

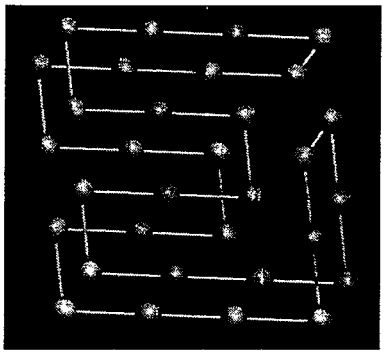
- Translational invariance the in 3,4,5,6 directions guarantees the 2-d $l_1 \times l_2$ curve is replicated.

- Orthogonality guarantees the links exist to make up remapped cartesian grid.

Keep on folding!

m_3 Orthogonal to both m_1 and $m_2 \Rightarrow$ iterate.

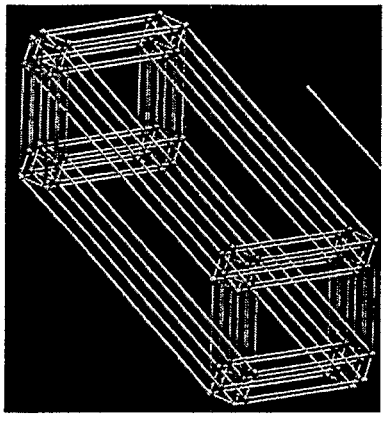
$(l_1 \times l_2 \times l_3) \cdot l_4 \cdot l_5 \cdot l_6 \cdot 1 \cdot 1$



$4 \cdot 4 \cdot 2 \longrightarrow 32$

m_4 and m_5 orthogonal to m_1 and m_2

$(l_1 \times l_2) \cdot (l_3 \times l_4) \cdot (l_5 \times l_6) \cdot 1 \cdot 1 \cdot 1$



2^6 motherboard $\longrightarrow 4 \cdot 4 \cdot 4$

QCDOC Front End External Interfaces & Services

David R. Stampf

Information Technology Division

Brookhaven National Laboratory

The QCDOC computer is substantially different in character from its predecessor the QCDSP. Changes include more processors, more memory/processor, faster processors and what we hope to be a growing user community with interests that diverge from those of the QCD/Lattice Gauge community. This will require computer access that is both more capable and more robust than what exists for the QCDSP machine. This presentation describe the external interface that will be available to the QCDOC users and it's internal architecture.

The QCDOC computer will not have a full fledged operating system running on it with provisions for user commands and interaction. It will be strictly an applications platform and all of the traditional operating system services will instead be placed on a front end computer that will be responsible for providing interactive capabilities, Grid protocols, Web access and an entry point for program access (e.g. Qcsh) to the QCDOC. We are currently evaluating front end systems for this purpose. The main requirements are that it support 8-10 Gigabit networks connections to the QCDOC and that it provide a reasonable platform for O/S work. (Good thread support, a good platform for Grid work, Apache/Tomcat web services, etc.) This will probably be satisfied by a modern Unix workstation.

Internal to this front end machine is a collection of software modules that has been named the "qdaemon". The qdaemon maintains three critical data structures – a list of Services, a list of Clients and an "in-use" table that protects objects in a multi-threaded environment. When an external user connects to the front end, they will first deal with acceptance and validation modules that will authorize access to the computer and grant certain privileges. When this is done, the new client is added to the Client table and is there-after reachable by any other module in the system. A Client Manager and Proxy object is then used to deal with the user. This object will handle negotiations with the user based upon the type of access (interactive, web, grid, etc.). Eventually, the user will need access to some sort of service. Services are available from the Service table by name. The user process will deal with the service for as long as necessary, and when the interaction is complete, will fall back to the client manager and proxy.

Once clients and services are in the table, they are available to all system components. We see the extension of the concept of a client to more abstract ideas such as "the monitor client", or the "system health watch client" that are really proxies for services that the QCDOC will need. In addition, Services may also be extended to include Log services and system status services that are more for the use of the QCDOC rather than users.

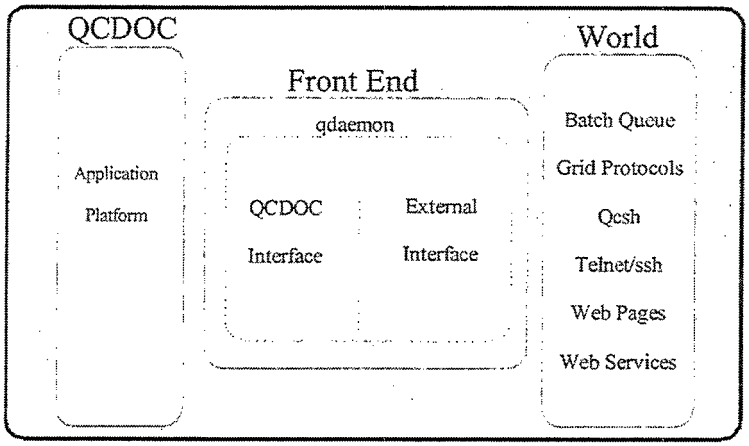
QCDOC Front End
External Interfaces & Services

Peter Boyle & Dave Stampf

Goals

- Transform the QCD user interface to QCDOC framework.
- Maintain expert level access
- Provide means of getting more data to more nodes and at faster speeds
- Provide means to share QCDOC among many users with diverse interests
- To minimize the O/S work at QCDOC nodes
- Provide Operational Interface

High Level View



Brookhaven Science Associates
U.S. Department of Energy

BROOKHAVEN
NATIONAL LABORATORY

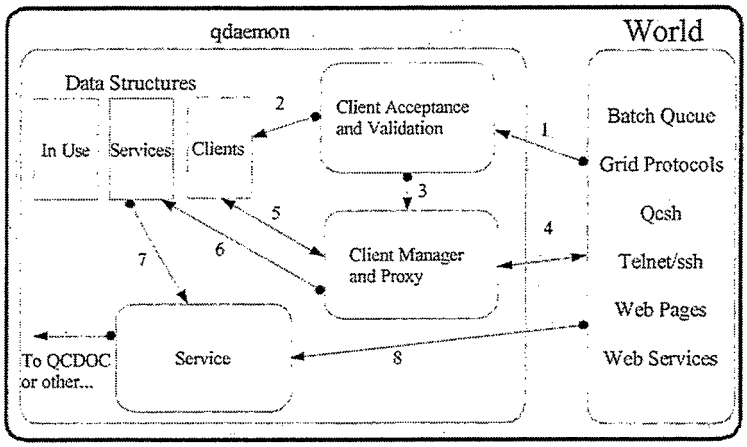
Front End Hardware & Software

- Modern workstation w/ 8-10 Gigabit network links to QCDOC
- Garden variety Unix system with excellent thread performance (most custom software is multi-threaded C++)
- Web Server (Apache/Tomcat) capable of basic http service and "web services"
- Must participate on "Grid" as QCDOC surrogate
- Support for PBS or similar queuing system

Brookhaven Science Associates
U.S. Department of Energy

BROOKHAVEN
NATIONAL LABORATORY

Software Architecture Map



Some Comments

- An object in-use table is used everywhere to protect objects in a multi-threaded environment
- Once a client/service is attached, they are available to all system components by name.
  - e.g. well known clients ("monitor") & services ("log")
- Services may communicate with QCDOC, but may also provide services to QCDOC or front end
- Clients and Services are abstract classes (interfaces) to provide design point not constraints

Error Report Handling

- Services are not limited to modules that deal with the QCDOC!
  - Log Service can be accessed by any other service or any other software module. (Test with MPI?)
  - Log Service will take error reports in an XML format (or a surrogate will format more random reports in an XML format) and send off to disk and/or monitor and/or database
  - Web servers and other internal Services will be able to access and analyze the error reports

Progress

- Framework is recently up and running
- No work yet on web/grid/PBS (however, it reuses most of the framework & other members of my group are capable of help here)
- Need experience to dictate services but we can easily create these from the abstract class
- Need to expand the DTDs for error reports & provide quick summary reports

Robert Bennett
Emerging Software Technology Group
Information Technology Division
Brookhaven National Laboratory

Talk title: MPI on QCDOC

This talk described the functionality and performance issues of implementing MPI, the standard message passing interface and library for high performance computing platforms, on the QCDOC machine.

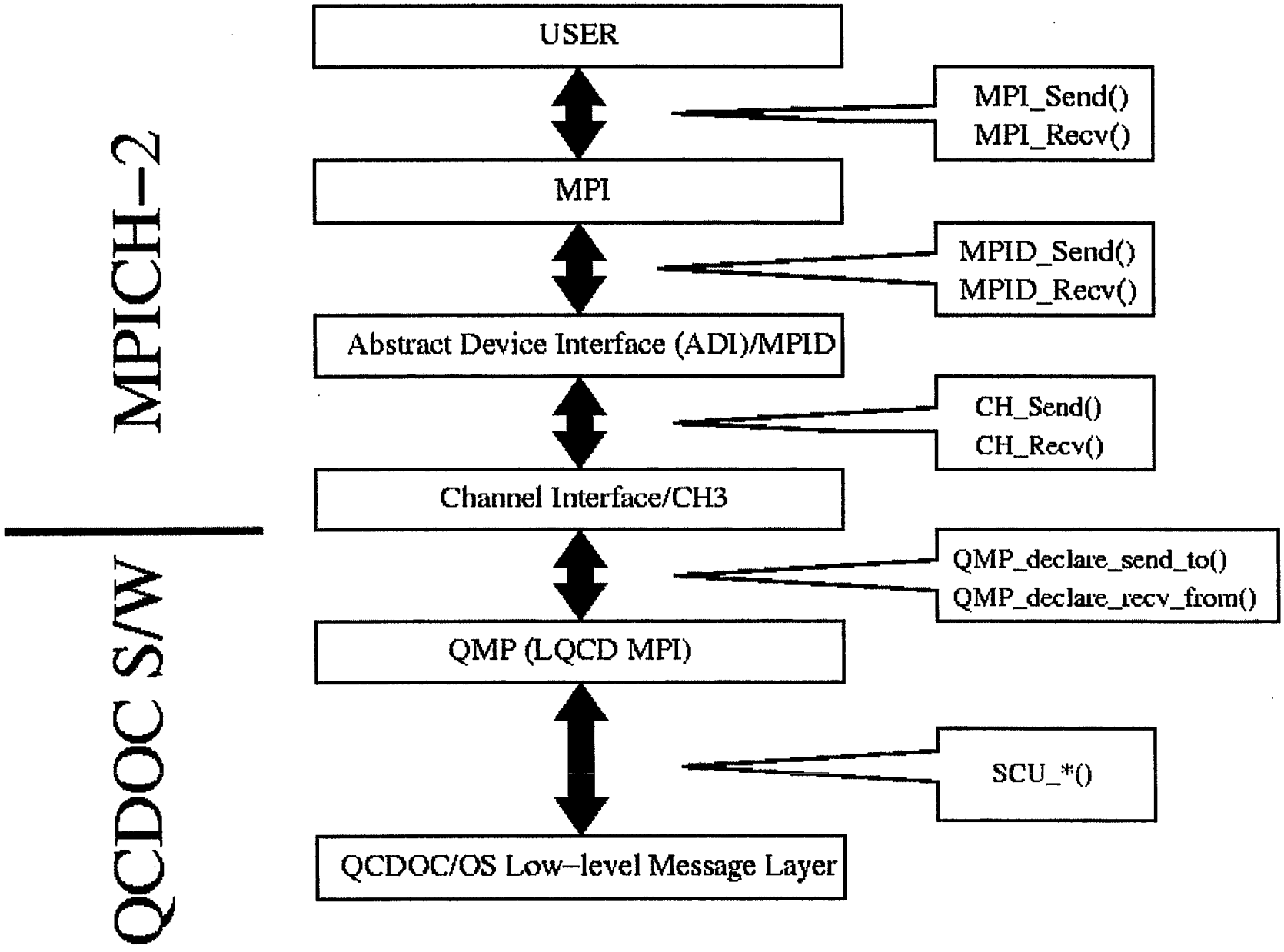
Objective

- Implement the core communication functionality of MPI on QCDOC
  - Allow MPI codes to run
  - Good performance for nearest-neighbor communication
  - Provide functionality for arbitrary communication
- Approach: Similar to MPI on BG/L [Gropp]

SciDAC QMP [Jung]

- Supports highly regular grid communication
  - Has
    - Point to point, non-blocking operations
    - Reduction operations (min/max, etc.)
    - Collective: Broadcast/Barrier
  - Doesn't have:
    - Message tagging & typing
    - Gather/Scatter collective operations + variants
- Handles message routing, packetization, etc.

MPI on QCDOC (ala MPI on BG/L)



MPI on QCDOC: Implementation Issues

- General message routing -> QMP
- Large buffer management -> QMP
- Handshaking for synchronization -> Channel/QMP/QCDOC-OS
- Packetization & queueing -> QMP
- Wildcard source (MPI_ANY_SOURCE) -> Channel/QMP/QCDOC-OS
- Message progress -> QMP
  - No threading, no second processor

Observations

- MPI point to point communication -> QMP
  - Channel: message tagging & synchronization
- MPI reduction operations -> QMP:
  - Store & forward h/w: Full volume & subdimensional subvolume
  - Channel: Arbitrary subvolumes
- MPI collective operations -> Channel/QMP

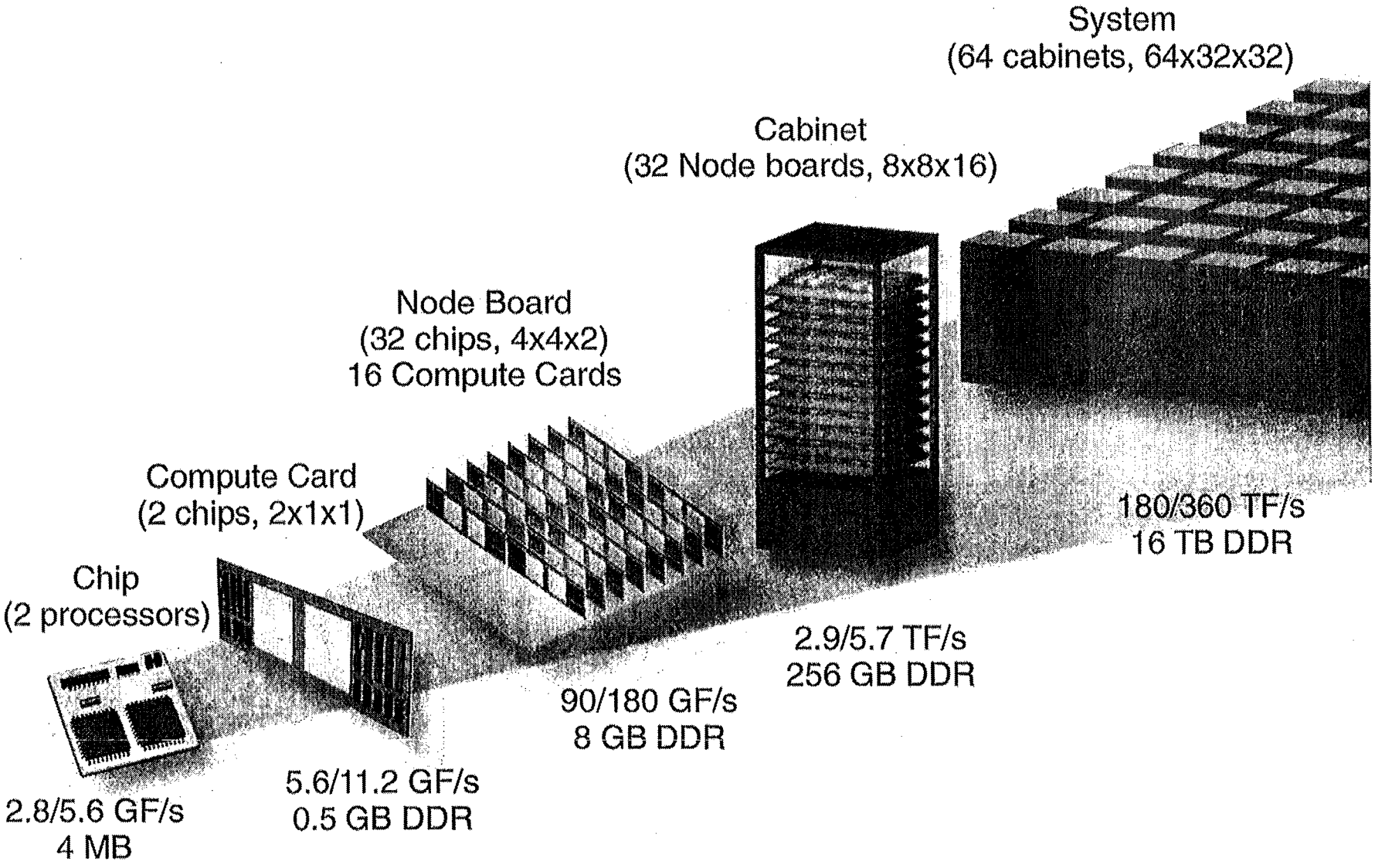
BlueGene/L Hardware Overview

Dong Chen
IBM T.J. Watson Research Center

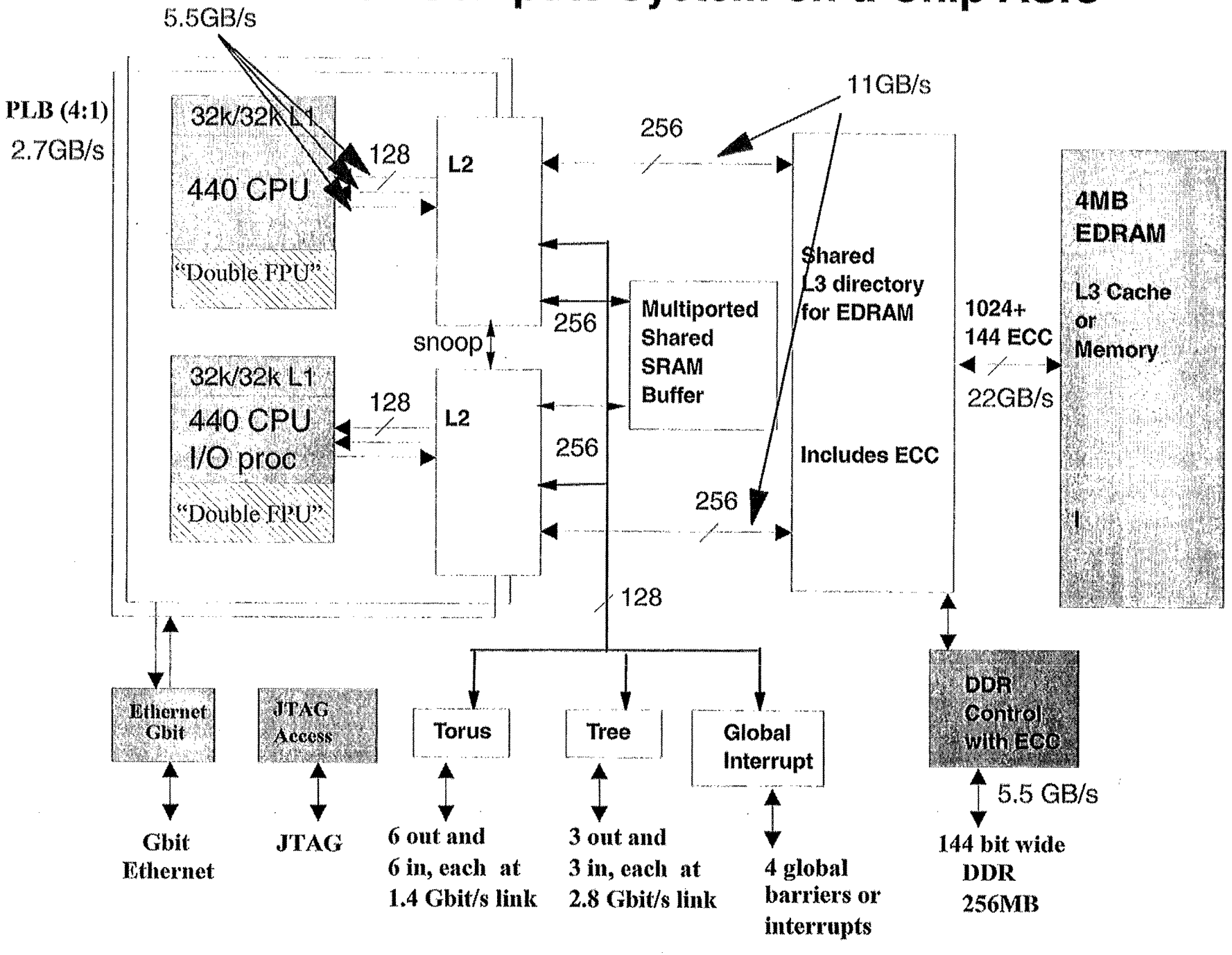
BlueGene/L is a massively parallel supercomputer builds on embedded System-On-a-Chip (SOC) technology. Each computing node consists of a single compute ASIC plus 256 MB of external memory. The compute ASIC integrates two 700 MHz PowerPC 440 integer CPU cores, two 2.8 Gflops floating point units, 4 MB of embedded DRAM as cache, a memory controller for external memory, six 1.4 Gbit/s bi-directional ports for a 3-dimensional torus network connection, three 2.8 Gbit/s bi-directional ports for connecting to a global tree network and a Gigabit Ethernet for I/O.

65,536 of such nodes are connected into a 3-d torus with a geometry of 32x32x64. The total peak performance of the system is 360 Teraflops and the total amount of memory is 16 TeraBytes.

Blue Gene/L

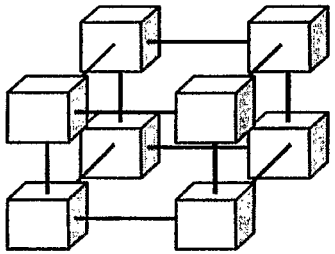

System
(64 cabinets, 64x32x32)
Cabinet
(32 Node boards, 8x8x16)
Node Board
(32 chips, 4x4x2)
16 Compute Cards
Compute Card
(2 chips, 2x1x1)
Chip
(2 processors)
2.8/5.6 GF/s
4 MB
5.6/11.2 GF/s
0.5 GB DDR
90/180 GF/s
8 GB DDR
2.9/5.7 TF/s
256 GB DDR
180/360 TF/s
16 TB DDR

BlueGene/L Compute System-on-a-Chip ASIC



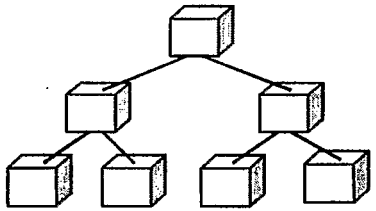
5.6GF peak node

BlueGene/L - Five Independent Networks



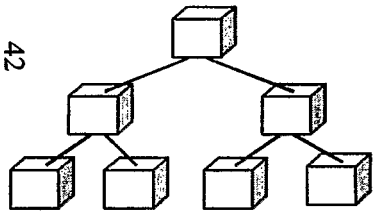
3 Dimensional Torus

- Point-to-point



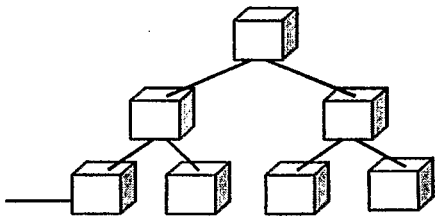
Global Tree

- Global Operations



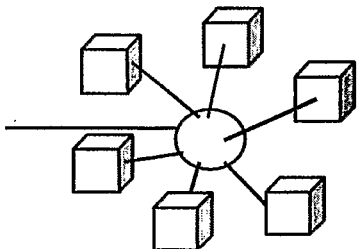
Global Barriers and Interrupts

- Low Latency Barriers and Interrupts



Gbit Ethernet

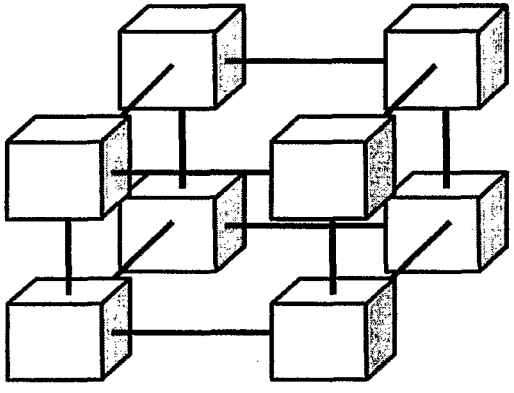
- File I/O and Host Interface

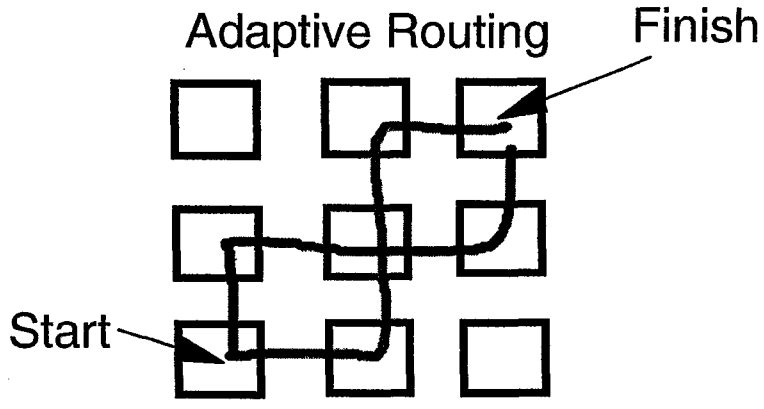


Control Network

- Boot, Monitoring and Diagnostics

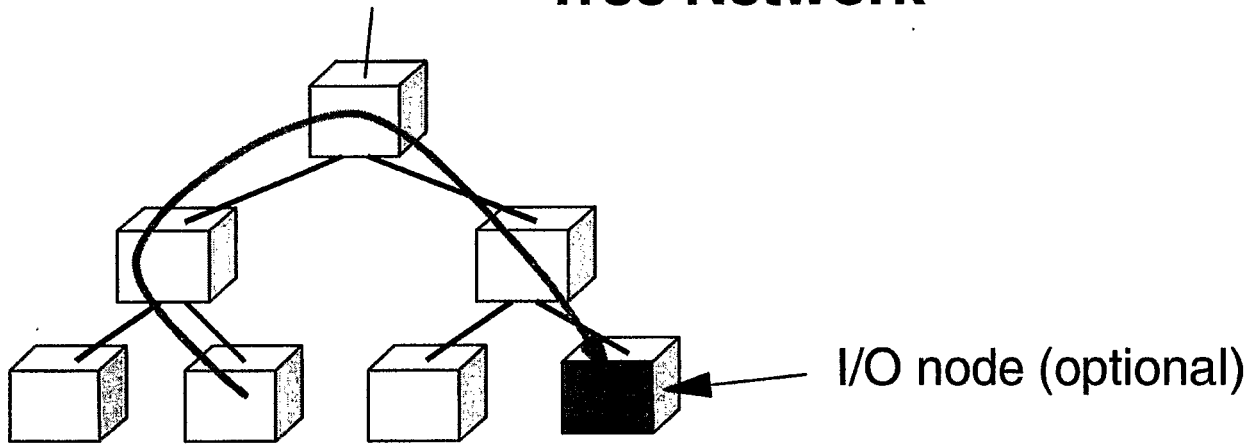
Three-dimensional Torus Network





- 32x32x64 connectivity
- Backbone for one-to-one and one-to-some communications
- 1.4 Gb/s bi-directional bandwidth in all 6 directions (Total 2.1 GB/s/node)
- 64k * 6 * 1.4Gb/s = 68 TB/s total torus bandwidth
- 4 * 32 *32 * 1.4Gb/s = 5.6 Tb/s Bisectional Bandwidth
- Worst case hardware latency through node ~ 69nsec
- Virtual cut-through routing with multipacket buffering on collision
  - Minimal
  - Adaptive
  - Deadlock Free
- Class Routing Capability (Deadlock-free Hardware Multicast)
  - Packets can be deposited along route to specified destination.
  - Allows for efficient one to many in some instances
- Active messages allows for fast transposes as required in FFTs.
- Independent on-chip network interfaces enable concurrent access.

Tree Network



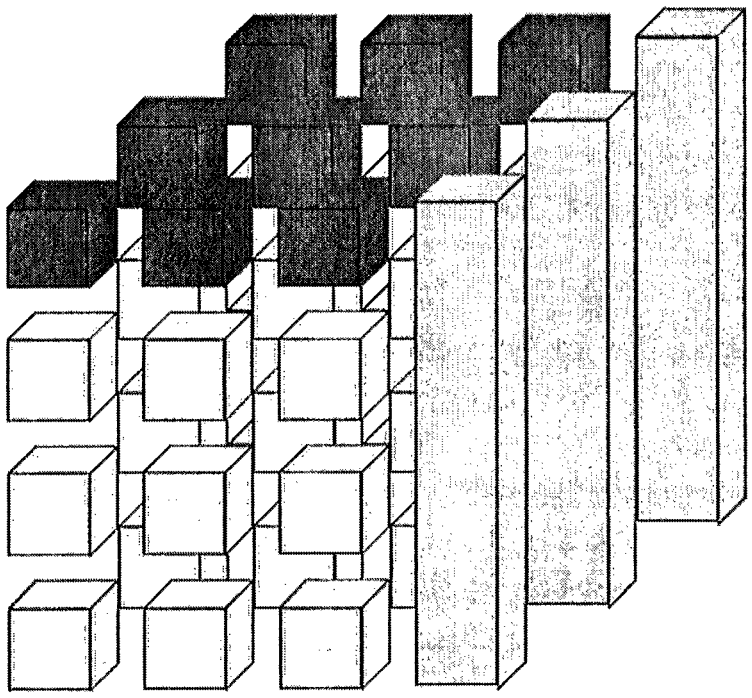
- High Bandwidth one-to-all
  - 2.8Gb/s to all 64k nodes
  - 68TB/s aggregate bandwidth
- Arithmetic operations implemented in tree
  - Integer/ Floating Point Maximum/Minimum
  - Integer addition/subtract, bitwise logical operations
- Latency of tree less than 2.5usec to top, additional 2.5usec to broadcast to all
- Global sum over 64k in less than 2.5 usec (to top of tree)
- Used for disk/host funnel in/out of I/O nodes.
- Minimal impact on cabling
- Partitioned with Torus boundaries
- Flexible local routing table
- Used as Point-to-point for File I/O and Host communications

BlueGene/L System Software Overview

José E. Moreira
IBM Thomas J. Watson Research Center
Yorktown Heights NY 10598-0218

With 65,536 compute nodes and 1,024 I/O nodes, BlueGene/L creates new challenges in scalability of system software services. In particular, system control services, job management services, I/O services, and communication services have to be designed to scale to those numbers. We solve those problems in BlueGene/L by organizing the system hierarchically. Application programs run and communicate exclusively on the compute nodes, which form the application volume of the machine. The machine interacts to the outside world for I/O and job management through the I/O nodes, which form the operational service of the machine. Each I/O node is the head of a processing set that comprises itself and 64 compute nodes. Finally, system control is performed by one or more service nodes, which form the control surface of the machine. Each I/O node runs a full Linux operating system image, while the compute nodes run a lightweight kernel that supports a single user process per compute node. For complex operations, like I/O, a compute process extends from the compute node into the I/O node. Application-level communication is directly supported at those compute nodes and requires no kernel intervention. The communication services are organized in three-layers: An active packets layer maps directly to the hardware, and supports operations on packets of at most 256 bytes. An active messages layer is built on top of the packets layer and supports operations on messages of arbitrary size. MPI is implemented on top of the active messages layer and is intended to be the primary communication mechanism for application programmers.

BG/L software architecture



- User applications execute exclusively on the compute nodes and only see the application volume as exposed by user level APIs
- The outside world interacts only with the I/O nodes and processing sets (I/O node + compute nodes) they represent, through the operational surface - functionally, the machine behaves as a cluster of I/O nodes
- Internally, the machine is controlled through the service nodes in the control surface - goal is to hide this surface as much as possible

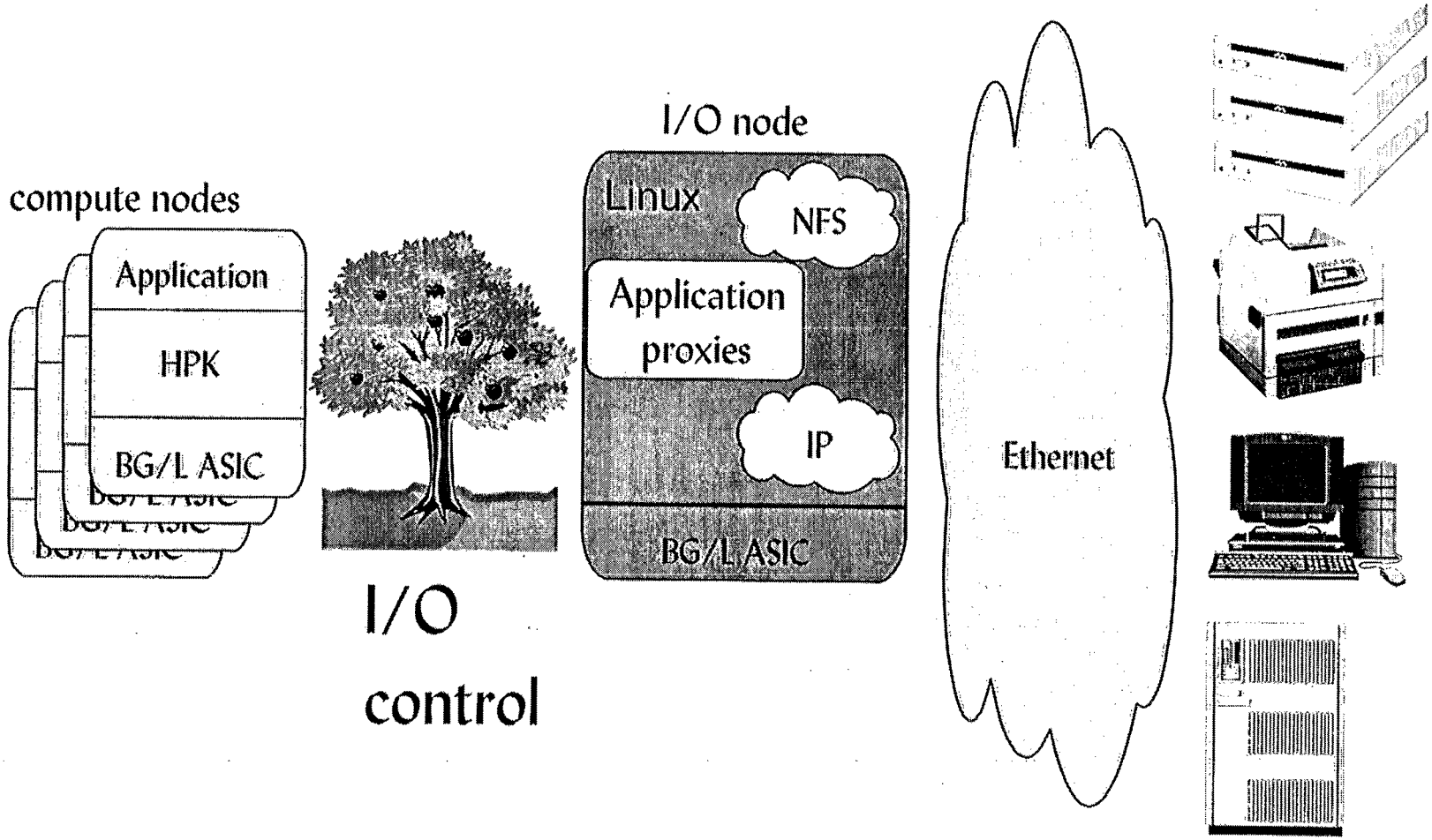
BG/L software architecture rationale

- We view the system as a cluster of 1024 I/O nodes, thus reducing a 65,536-node machine to a manageable size
- From a system perspective, user processes "live" in the I/O nodes:
  - process management (start, monitor, kill)
  - process debugging
  - authentication, authorization
  - system management daemons (LoadLeveler, xCAT, MPI)
  - traditional Linux operating system
- User processes actually execute on compute nodes
  - simple single-process (two threads) kernel on compute nodes
  - one processor/thread provides fast, predictable execution
  - user process extends into I/O node for complex operations
- Application model is a collection of private-memory processes communicating through messages

Programming models for compute nodes

- Heater mode: CPU 0 does all the work for computation and communication, while CPU 1 spins idle
  - intended primarily for debugging mode of system
  - some applications may actually benefit!
  - reduces power consumption
- Coprocessor mode: CPU 0 executes user application while CPU 1 executes coroutines
  - requires coordination between CPUs, which is handled in libraries
  - coroutine library for handling communications: preferred mode of operation for communication-intensive codes
  - can be used with other coroutine libraries, e.g., computation coprocessor
- Virtual node mode: Both CPUs execute the user application and also handle communication
  - attractive for compute-bound applications
  - separate processes (communicating via MPI) execute on CPU 0 and CPU 1
  - two single-threaded processes per compute node

I/O node role in the system



Communication infrastructure for BG/L

- Three layers of communication libraries:
  - SPI active packets layer - directly maps to hardware
  - SPI active messages layer - abstraction layer
  - MPI - for end user application
- Active packets and active messages are self-describing entities that cause actions to be executed on the receiving node
  - packets have a maximum length determined by architecture (256 bytes)
  - messages can be much longer and are broken down into packets
  - can be used by applications, but are intended more for library developers
  - one-sided communication, which can be acted upon as soon as it is received by the target node
- MPI is based on active messages layer
  - first step was a simple port of MPICH based on active messages
  - we are now enhancing MPI will knowledge of machine topology
  - we will use the BG/L tree for global reductions and broadcasts

Microsecond Simulations for MD and Related Algorithms

Yuefan Deng, Department of Applied Mathematics and Statistics,
University at Stony Brook, Stony Brook, NY

James Glimm, Department of Applied Mathematics and Statistics,
University at Stony Brook, Stony Brook, NY
and
Center for Data Intensive Computing,
Brookhaven National Laboratory, Upton, NY

James Davenport, Center for Data Intensive Computing,
Brookhaven National Laboratory, Upton, NY

We propose to simulate 10 or more microseconds of physical time for 100,000 particles interacting with short and long-range (Coloumb) forces, using 8000 nodes of the QCDOC. This estimate is based on the use of the Ewald algorithm, an analysis of the Ewald algorithm, and detailed analysis of the allowed cutoffs in relation to errors. We propose to take femtosecond time steps for the short-range forces, and to remove these short-range components from the Ewald sum, so that longer time steps for this more expensive part of the algorithm will be allowed.

The estimates are based on published performance figures for the QCDOC in terms of floating point performance and network communication performance. We plan to keep all data in level 2 cache (on chip memory), so that there will be no level 2 cache misses, and thus a high single processor performance should be attained. Conservatively we are estimating 30% of peak performance in our plans. The network is characterized by a latency and a bandwidth parameter for each of the communication channels. We have estimated the influence of these parameters on the required global communication patterns. To do this it was necessary to plan the sequence of messages needed for an allgather communication. We found a nearly 50% utilization of the available hardware bandwidth capability of the network. Latency is not close to being a limiting factor in this model of the simulations.

The Ewald algorithm will be useful for molecular dynamics simulations, stochastic molecular dynamics simulations (randomly perturbed to emulate a heat bath) and Monte Carlo simulations.

Microsecond MD Simulations on QCDO

Jim Davenport, Yuefan Deng, and James Glimm
CDIC, Brookhaven National Laboratory and Stony Brook University

- Significance of results
- Timing models and estimates
  - Relevant QCDOC hardware model
  - MD communication algorithms
  - MD computation algorithms
  - MD performance models
- Generalizations of QCDOC Applications
  - Global Communication model
  - Applications: Quantum Chemistry and FFT

Molecular Dynamics

- Strong scaling
  - 14 particles per processor (10^5 total)
  - 15 fold improvement over best methods for a Beowulf system
- Main Result
  - 1 – 10 microseconds MD time with 10^4 processors
  - MD applications may scale to 10^5 processors
  - Algorithm is computation, not communication, bound

Timing Models and Estimates

- Analysis of Ewald algorithm
- Analysis of QCDOC hardware
- MD performance estimates as a function of
  - Accuracy of computation
  - Number of particles
  - Number of processors
- Conclusion: timing is computationally limited with 14 particles per processor (total 10^5)
  - Strong scaling
  - 1 - 10 microseconds of physical time simulated

Ewald algorithm

Allowed error determines kmax and rmax, and number of operations required. Ewald parameter α adjusts balance between kmax and rmax and achieves a minimum total number of operations.

A theoretical analysis determines large kmax, rmax asymptotes. Numerical experiments determine exact relation and exact operation counts.

Assume 1/3 efficiency (conservatively) relative to peak performance with data in L2 cache.

Above determines speed per time step.

Assume slow time step (local forces removed) of 10^{-14} sec. and a 10^{-15} time step for local forces

Result is estimate on simulation: 1 - 10 microseconds

Timing: Summary

- Short range part of Coulomb is solved separately, so that longer time steps (10 fsec) are allowed for Ewald terms
- Per step timing estimates:
  - $t_{computation} = 10^{-2}$ sec
  - $t_{Allgather\ bandwidth} = 7.5 \times 10^{-3}$ sec
  - $t_{Allgather\ latency} = 9 \times 10^{-6}$ sec
- Present analysis does not address finite size of communication buffers. Latency will rise as this is added

Radiation-Hydrodynamic Simulations of Core Collapse Supernovae on Terascale Platforms

F. Douglas Swesty
Dept. of Physics & Astronomy
SUNY Stony Brook

Radiation-hydrodynamic simulations of core collapse supernovae provide a demanding testbed problem for emerging ultrascale parallel computing platforms. These simulations require the explicit time-evolution of 3-D hydrodynamic equations (which describe the flow of material in the collapsing star) along with the implicit time-evolution of multiple sets of radiation transport equations (which describe the flow of neutrinos through the star). Since the distribution function for the neutrinos is the unknown, the equations span not only the three spatial dimensions but also the three momentum dimensions. The transport equations are implicitly discretized over this space yielding a large sparse non-linear system of equations. The computational cost of the simulations is dominated by the solution of these coupled sets of non-linear equations. These sparse systems are solved using a combination of Krylov subspace iteration along with Newton-Raphson iteration in the form of Newton-Krylov methods.

The parallelism of the problem is derived from spatial domain decomposition of the global spatial domain into tiles (2D) or cubes (3D). Under such a decomposition the implementation of Newton-Krylov methods is straightforward and can be accomplished via a combination of embarrassingly parallel BLAS operations, a matrix-vector multiply (requiring only nearest-neighbor communication), and vector inner products (requiring global reduction operations). It is the global reduction operations that present the greatest challenge to scalability to thousands of processors. For this reason innovative architectures, such as QCDOC and BlueGene, which can provide fast global reduction operations represent a potential path towards the next generation of core collapse supernova models.

Supernova Simulation Components

- Eulerian Hydrodynamics (explicit algorithms)
  - Newtonian hydrodynamics
  - General relativistic hydrodynamics

- Neutrino transport (implicit algorithms)
  - Dominates memory and I/O
  - Multi-group Boltzmann transport
    - Sparse systems algorithms (matrix-free)
  - Multi-group Flux-limited Diffusion
    - Sparse systems algorithms

- Nuclear & particle microphysics
  - Equation of state (EOS) model
  - Opacities & Neutrino absorption/emission rates
  - Reactive flow nuclear chemistry

F. D. Swesty

STONY BROOK

http://www.astro.sunysb.edu/dswesty
dswesty@mail.astro.sunysb.edu

Newtonian Hydrodynamics

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$ **Continuity Equation**

$$\frac{\partial E}{\partial t} + \nabla \cdot (E\mathbf{v}) + P\nabla \cdot \mathbf{v} = \mathsf{S}$$ **Gas Energy Conservation**

$$\frac{\partial \rho v_i}{\partial t} + \nabla \cdot (\rho v_i \mathbf{v}) + (\nabla P)_i + \rho (\nabla \Phi)_i = \mathsf{A}_i$$ **Gas Momentum Conservation**

-Discretize variables on a 1-D, 2-D, or 3-D spatial mesh

-Solution via explicit finite difference or finite volume techniques

-Courant-Friedrichs-Lewy stability criterion on timestep size: $\Delta t < \frac{1}{2}\frac{\Delta x}{c_s}$

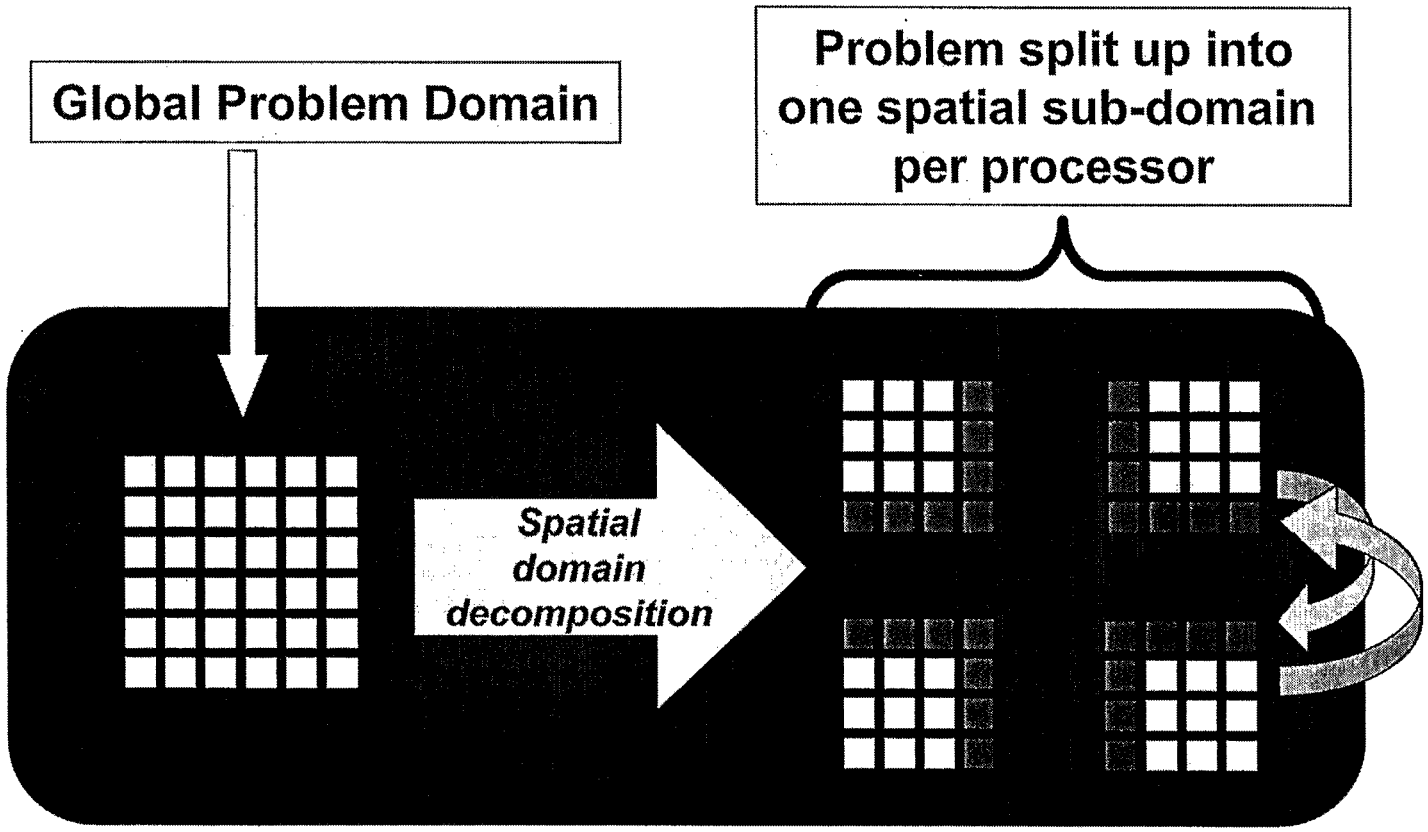
F. D. Swesty

STONY BROOK

QCDOC-Blue Gene/L Workshop BNL 2/28/03

http://www.astro.sunysb.edu/dswesty
dswesty@mail.astro.sunysb.edu

Parallel Implementation



F. D. Swesty
STONY BROOK

http://www.astro.sunysb.edu/dswesty
dswesty@mail.astro.sunysb.edu

O(v/c) Multigroup Neutrino Energy Approximation

-These equations are have 3 spatial + 1 spectral dimension
- Must solve 1 pair of equations for each neutrino flavor

Neutrinos

$$\frac{\partial E_\epsilon}{\partial t} + \nabla\cdot(E_\epsilon \mathbf{v}) + \nabla\cdot\mathbf{F}_\epsilon - \mathsf{P}_\epsilon : \nabla\mathbf{v} + \frac{2\mathbf{a}}{c^2}\cdot\mathbf{F}_\epsilon - \nabla\mathbf{v} : \frac{\partial}{\partial\epsilon}(\epsilon\mathsf{P}_\epsilon) - \frac{\mathbf{a}}{c^2}\cdot\frac{\partial}{\partial\epsilon}(\epsilon\mathbf{F}_\epsilon)$$
$$= \mathsf{S}(\epsilon) \equiv S_\epsilon\left(1-\frac{\alpha}{\epsilon^3}E_\epsilon\right) - c\kappa^a_\epsilon E_\epsilon + \left(1-\frac{\alpha}{\epsilon^3}E_\epsilon\right)c\int d\epsilon'\kappa^s(\epsilon',\epsilon)E_{\epsilon'}$$
$$-E_\epsilon c\int d\epsilon'\kappa^s(\epsilon,\epsilon')\left(1-\frac{\alpha}{\epsilon'^3}E_{\epsilon'}\right) + \left(1-\frac{\alpha}{\epsilon^3}E_\epsilon\right)\epsilon\int d\epsilon' G(\epsilon,\epsilon')\left(1-\frac{\alpha}{\epsilon'^3}\bar{E}_{\epsilon'}\right)$$

Anti-Neutrinos

$$\frac{\partial \bar{E}_\epsilon}{\partial t} + \nabla\cdot\left(\bar{E}_\epsilon \mathbf{v}\right) + \nabla\cdot\bar{\mathbf{F}}_\epsilon - \bar{\mathsf{P}}_\epsilon : \nabla\mathbf{v} + \frac{2\mathbf{a}}{c^2}\cdot\bar{\mathbf{F}}_\epsilon - \nabla\mathbf{v} : \frac{\partial}{\partial\epsilon}\left(\epsilon\bar{\mathsf{P}}_\epsilon\right) - \frac{\mathbf{a}}{c^2}\cdot\frac{\partial}{\partial\epsilon}\left(\epsilon\bar{\mathbf{F}}_\epsilon\right)$$
$$= \bar{\mathsf{S}}(\epsilon) \equiv \bar{S}_\epsilon\left(1-\frac{\alpha}{\epsilon^3}E_\epsilon\right) - c\bar{\kappa}^a_\epsilon\bar{E}_\epsilon + \left(1-\frac{\alpha}{\epsilon^3}\bar{E}_\epsilon\right)c\int d\epsilon'\bar{\kappa}^s(\epsilon',\epsilon)\bar{E}_{\epsilon'}$$
$$-\bar{E}_\epsilon c\int d\epsilon'\bar{\kappa}^s(\epsilon,\epsilon')\left(1-\frac{\alpha}{\epsilon'^3}\bar{E}_{\epsilon'}\right) + \left(1-\frac{\alpha}{\epsilon^3}\bar{E}_\epsilon\right)\epsilon\int d\epsilon' G(\epsilon',\epsilon)\left(1-\frac{\alpha}{\epsilon'^3}E_{\epsilon'}\right)$$

F. D. Swesty

STONY BROOK

QCDOC-Blue Gene/L Workshop BNL 2/28/03

http://www.astro.sunysb.edu/dswesty
dswesty@mail.astro.sunysb.edu

Platform Requirements for Radiation-Hydro Simulations on Terascale/Ultrascale Platforms

Experimental Platform

- -Compiler (F90 a plus but not required)
- -MPI
- -128 Mbytes/processor minimum; 256 Mbytes better

Production Platform

- -F90 a must!
- -MPI
- -256 Mybtes/processor minimum; 512 Mbytes better
- -Parallel I/O via MPI-I/O w/ parallel filesystem
- -Terabyte+ scratch space

Blue Gene Application Overview

Robert S. Germain
IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598

In December 1999, IBM Research announced a 5 year, $100M US, effort to build a petaflop scale supercomputer to attack problems such as protein folding. The applications effort within IBM has remained focused on the life sciences and on protein dynamics in particular throughout the evolution of the project. The scientific goal is to use large scale simulation studies to improve our understanding of biologically important processes, in particular the mechanisms behind protein folding. This involves some interesting challenges for the N-body simulation application that supports this scientific effort. In contrast to many scientific applications where scaling with respect to problem size is the key, protein dynamics studies require long time simulations and therefore scaling of execution rate with processor count for a fixed problem size is the metric of interest.

In order to accurately model the protein and peptide systems that we wish to study, an accurate treatment of long range electrostatic interactions is required (with periodic boundary conditions). We are exploring the use of the global tree network in BG/L to simplify the application logic and we are also sizing distributed memory 3D FFTs on the BG/L torus network. These theoretical sizings will help us prioritize our development efforts prior to the availability of BG/L hardware. There is also work taking place on sizing other computational biology applications such as finite difference models of the heart as well representative applications of interest outside of the life sciences.

Application Overview

R. S. Germain
rgermain@us.ibm.com
Biomolecular Dynamics
and Scalable Modeling
http://www.research.ibm.com/bluegene/

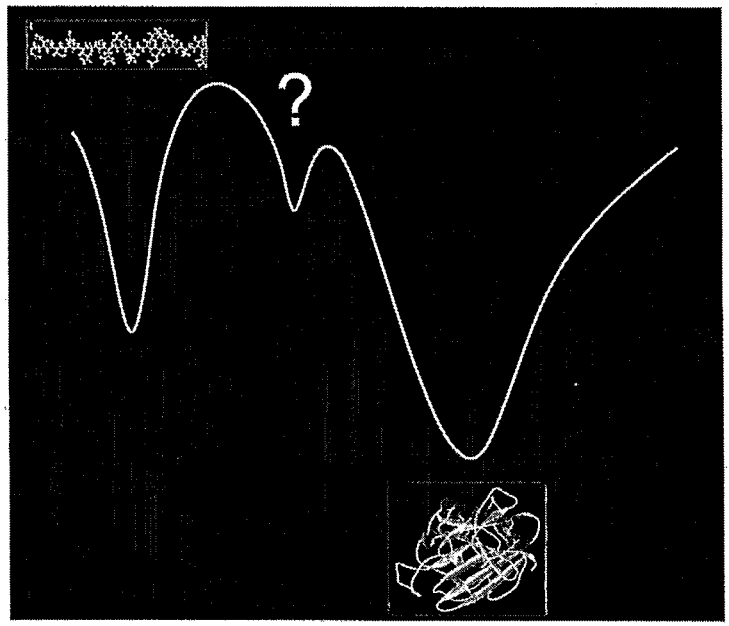
March 12, 2003

Outline

- Introduction
- Machine architecture (from application perspective)
- N-body simulation (molecular dynamics for biomolecular simulation)
- Finite difference simulation (heart model)
- Other application assessments

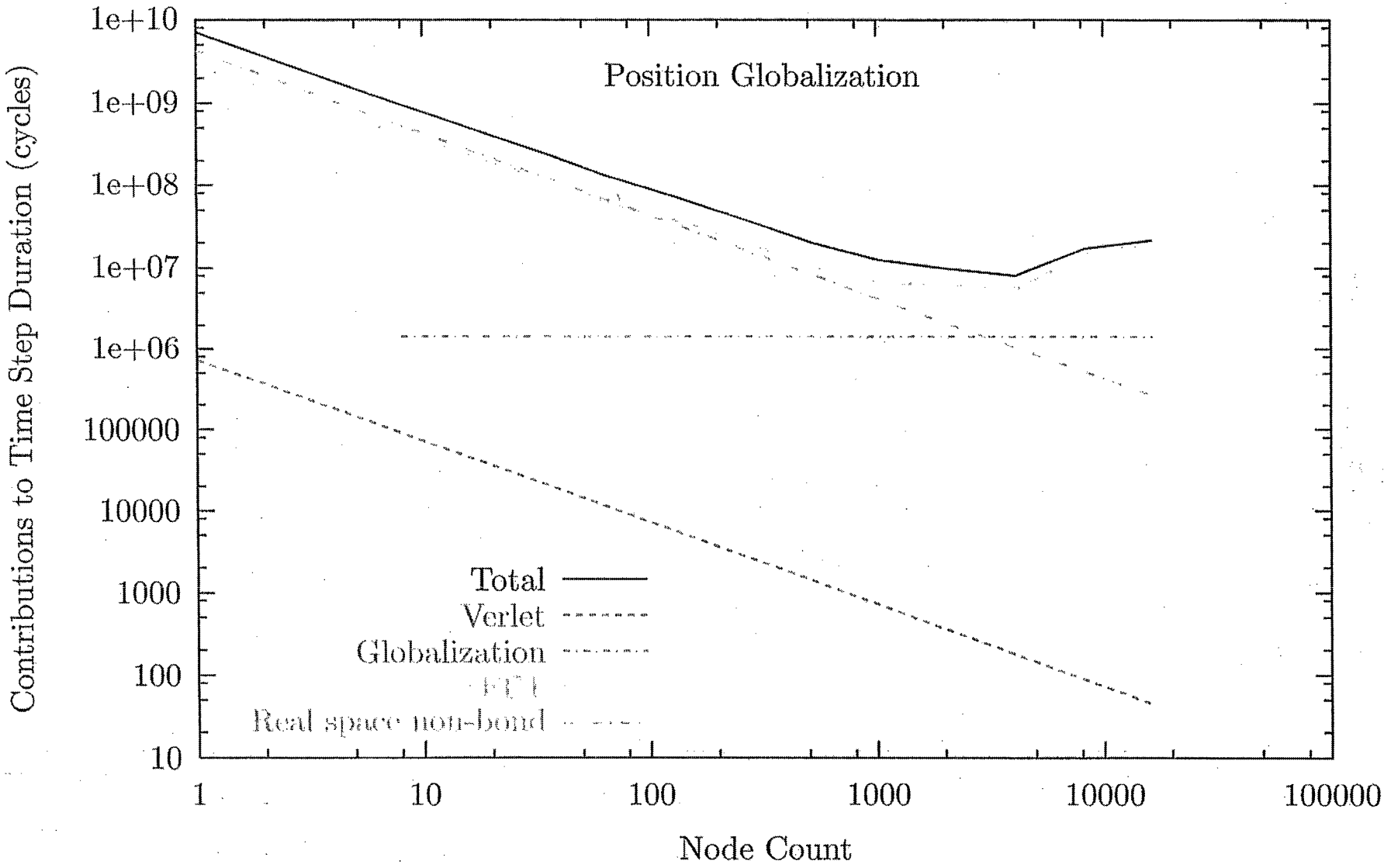
Protein Folding and Blue Gene

- Why does a protein fold?
  - Thermodynamics: Describe the intermediate structures, the energy, entropy, free energy landscape analysis along the "folding path", without interest in kinetics.
- How does it fold so quickly?
  - Kinetics: Describe the rates associated with phases of the folding process, transition mechanisms, and the time spent in various states along the pathway.
- What structure does it fold to?
  - Structure Prediction: Predict the folded structure

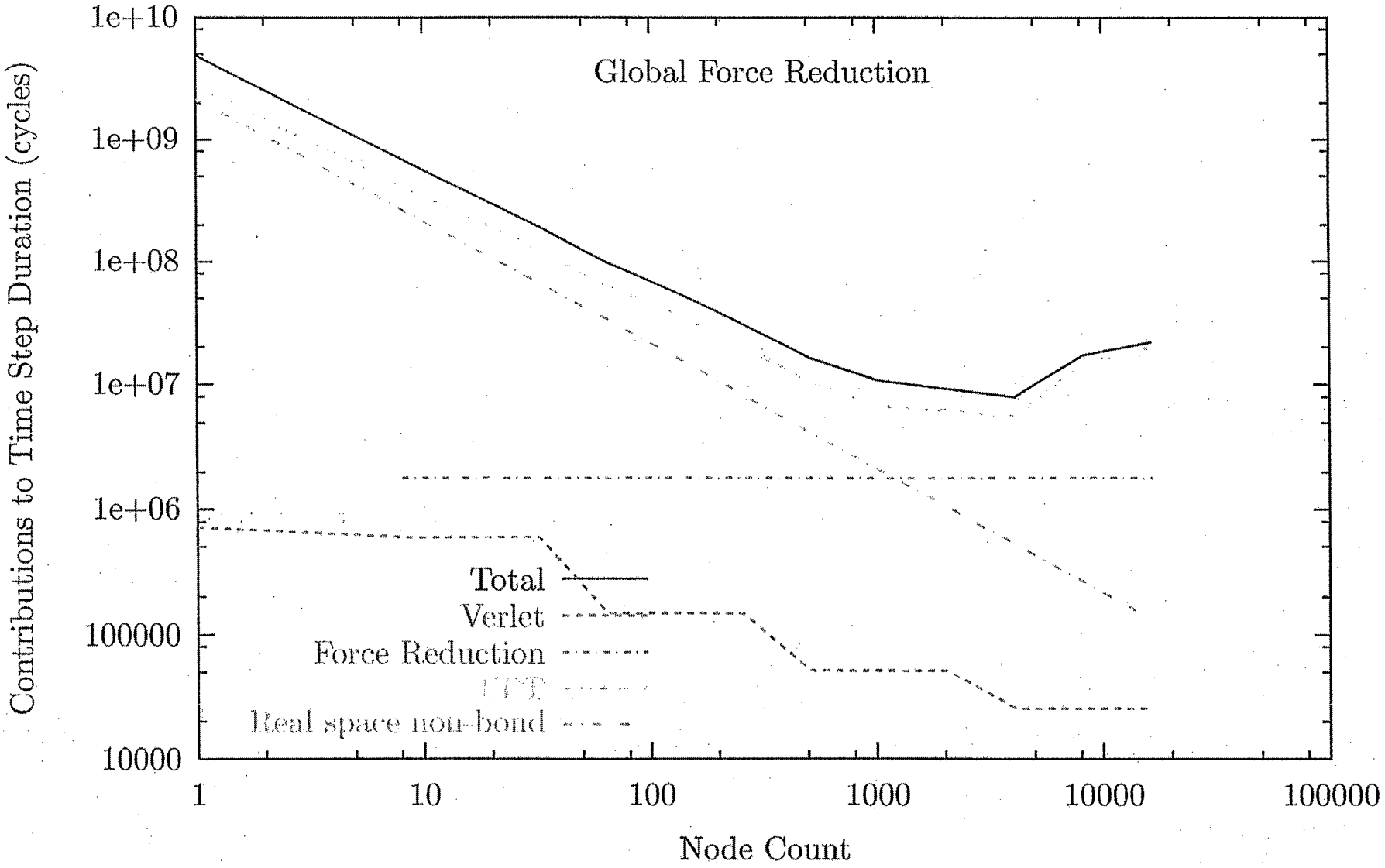


- Starting with simple systems, the Blue Gene science program will use high-quality thermodynamic and kinetic simulations to study the protein folding process.
- Classical molecular dynamics all-atom simulations with explicit solvent

Contributions to MD Time Step (Position Globalization)

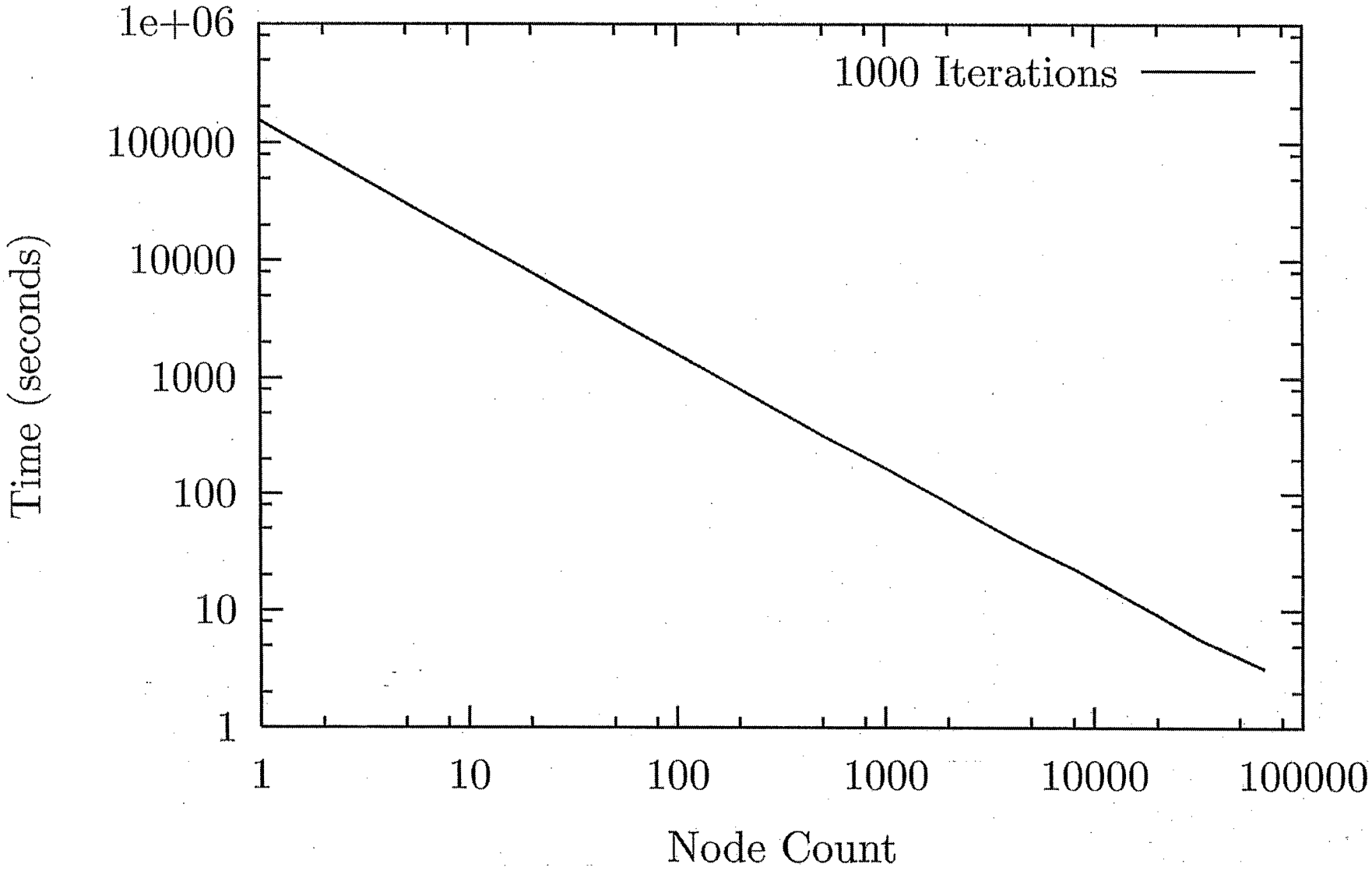


Contributions to MD Time Step (Global Force Reduction)



Finite Difference Code:

$1024 \times 1024 \times 1024$



Performance Stresspoints for Parallel Implicit PDEs

David E. Keyes[1]

Most progressive approaches for solving large sparse nonlinear systems arising from locally discretized PDEs on parallel computers—involving multilevel, operator-splitting, Newton, Krylov, and Schwarz algorithmic ideas—employ a relatively small set of basic operations on extensively reused parallel data structures. The evaluation of the implicitly discretized nonlinear vector-valued conservation law residual function, based on a local multicomponent PDE stencil, is basic to all approaches and consumes the majority of execution time after the solver itself is optimized. Other operations include fine-coarse intergrid transfers, sparse matrix-vector multiplications, processor-local recurrences (e.g., backsolves, relaxation sweeps), interprocessor exchanges for near-neighbor data dependencies, and global reductions.

Not surprisingly, these different operations may stress very different parts of a high-performance machine. The function evaluations, with their independent writes and high reuse of register data, are typically able to run at the highest levels of performance and are amenable to multithreading. Their main difficulty comes from EOS and other model-specific tasks that may be difficult to load-balance simultaneously with the algebraic phases. Sparse linear algebra routines, with low cache reuse, are typically memory bandwidth-limited. Synchronizing global reductions are ultimately bottlenecks in the limit of large network diameter and imperfect load balance. Intergrid operations may be limited by bulk interprocessor bandwidth, depending upon subdomain-to-processor mappings and network routing support. This talk presents computational experiments on existing high-end machines and conclusions of examining these issues with simple complexity models. It also seeks to motivate the suitability of QCDOC for the large class of DOE mission-relevant applications that is well described by PDEs, subject to possible memory size constraints. Results of an aerodynamics computation with a well-understood performance signature are naïvely extrapolated to QCDOC.

With respect to the most critical performance bottleneck of the ASCI platforms—bandwidth to main memory—QCDOC looks very promising. It also appears to have excellent potential for low latency global reductions, if required multilayered system software can be built over the native communication protocols in a low-overhead manner. Other issues that need to be addressed to clear the way for general purpose PDE computation on QCDOC fall into architectural, software, and algorithmic categories. Architecturally, local memory size and I/O rates may need attention. Production PDE codes need a full software environment, including scientific compilers, MPI-1 and MPI-IO, debuggers, etc. Algorithmic issues are fairly generic but are exacerbated by small memory per node. Locality-enhancing ordering, aggregation of horizontal and vertical transfers, and communication-hiding split transactions are critical. Cacheing lookup tables and restructuring for minimal integer overhead are also critical in low-memory situations.

[1]Richard F. Barry Professor of Mathematics & Statistics, Old Dominion University, Norfolk, VA 23529-0077; Acting Director, Institute for Scientific Computing Research, Lawrence Livermore National Lab, L-419, Livermore, CA 94551-9989. E-mail and web: dkeyes@odu.edu, http://www.math.odu.edu/~keyes.

Performance Stresspoints for Implicit PDE Simulations

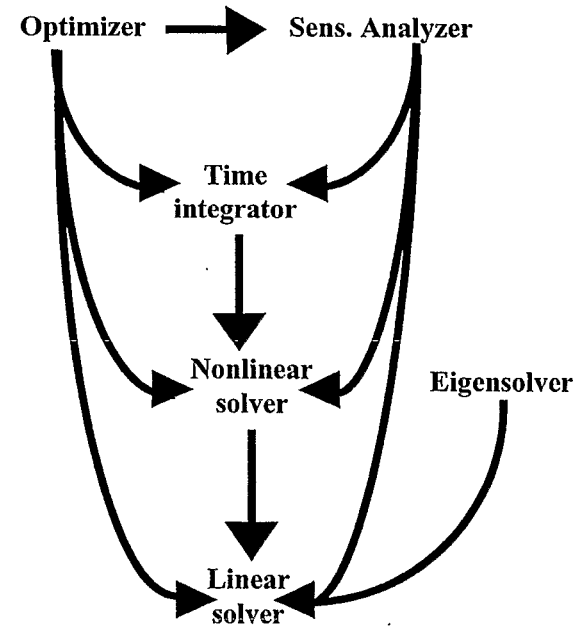
QCDOC Workshop
Brookhaven National Lab
28 February 2003

David Keyes
Center for Computational Science, Old Dominion University
&
Institute for Scientific Computing Research, Lawrence Livermore National Lab

www.math.odu.edu/~keyes/talks/qcdoc_2003.ppt
.../papers.html

Toolchain for PDE Solvers in TOPS* project

- **Design and implementation of "solvers"**
  - **Time integrators (w/ sens. anal.)** $f(\dot{x}, x, t, p) = 0$
  - **Nonlinear solvers (w/ sens. anal.)** $F(x, p) = 0$
  - **Constrained optimizers** $\min_u \phi(x,u)\ s.t.\ F(x,u) = 0, u \geq 0$
  - **Linear solvers** $Ax = b$
  - **Eigensolvers** $Ax = \lambda Bx$



- **Software integration**
- **Performance optimization**

→ **Indicates dependence**

*Terascale Optimal PDE Simulations: **www.tops-scidac.org**

TOPS Terascale Optimal PDE Simulations

Memory vs. work scaling for grid-based PDEs

- **For 3D problems, work is proportional to four-thirds power of memory, because**
  - **For equilibrium problems, work scales with problem size times number of iteration steps – roughly proportional to resolution in single spatial dimension (better for multilevel convergence-optimal methods)**
  - **For evolutionary problems, work scales with problems size times number of time steps – CFL arguments place latter on order of spatial resolution, as well**
- **Proportionality constant can be adjusted over a very wide range by both discretization (high-order implies more work per point and per memory transfer) and by algorithmic tuning**
- **If frequent time frames are to be captured, other resources -- disk capacity and I/O rates -- must both scale linearly with work, more stringently than for memory**

Primary PDE solution kernels*

- **Vertex-based loops**
  - **state vector and auxiliary vector updates**
- **Edge-based "stencil op" loops**
  - **residual evaluation**
  - **approximate Jacobian evaluation**
  - **Jacobian-vector product (often replaced with matrix-free form, involving residual evaluation)**
  - **intergrid transfer (coarse/fine) in multilevel methods**
- **Sparse, narrow-band recurrences**
  - **approximate factorization and back substitution**
  - **smoothing**
- **Vector inner products and norms**
  - **orthogonalization/conjugation**
  - **convergence progress and stability checks**

***assumes vertex-based; dual statements for cell-based**

Summary of observations for PDE codes

- **Processor scalability no problem, in principle, provided there is a sufficiently rich interconnection network (mesh/torus okay for PDEs)**
- **Memory latency no problem, in principle, with proper locality-based ordering**
- **Memory bandwidth is likely a *major* bottleneck**
- **Instruction scheduling *may* be a bottleneck in some physics kernels on some imbalanced processors, e.g., insufficient load/store units relative to bandwidth and FPUs**
- **Low frequency of floating point instructions is an algorithmic bottleneck intrinsic to unstructured problems, which requires (and has prospects for) algorithmic remedy**

Optimal schemes for Car-Parrinello based *ab initio molecular* dynamics on parallel architectures.

Mark E. Tuckerman

Dept. of Chemistry and Courant Institute of Mathematical Sciences,

New York University, New York, NY 10003

The field of *ab initio* molecular dynamics, in which finite temperature molecular dynamics trajectories are generated with forces obtained from density functional electronic structure calculations performed "on the fly", is a rapidly evolving and growing technology that allows chemical processes in condensed phases to be studied in an accurate and unbiased way. In this talk, Car-Parrinello approach to *ab initio* molecular dynamics is briefly described and several algorithms for implementing the method on parallel architectures are discussed. Much of the talk focuses on the use of plane-wave basis sets for expansion of the electronic orbitals, and two parallelization schemes are considered. The first is a hybrid scheme in which operations on individual electronic states and density operations on the real-space and reciprocal-space grids are distributed over processors. The second is one in which *all* grid-based calculations, both on the electronic states and on the density, are distributed. These schemes are shown to scale well up to 64-128 processors.

Next, the problem of achieving efficient scaling on massively parallel architectures is discussed. In particular, the use of localized basis sets as a means of eliminating dense all-to-all type communication is described. The particular scheme chosen is based on the use of so called discrete variable representations, which are highly localized functions at individual points of a quadrature mesh. This choice is shown to reduce the size of the grid by a full order of magnitude over plane waves. Finally, a new paradigm is discussed, which incorporates the charm++ utility of L. V. Kale and coworkers. Charm++ is a runtime parallel scheduler that dynamically assigns a given number of tasks to available processors in such a way as to achieve optimal load balancing. Charm++ also allows tasks to be migrated as the calculation or computational resources change. Preliminary results show good scaling for *ab initio* molecular dynamics on up to 512 processors. The problem of redesigning code in order to make use of the charm++ utility is also discussed.

Car-Parrinello molecular dynamics

- Alternative to explicit minimization or diagonalization
- Fictitious electron dynamics used to generate approximate minimized electronic distribution at each nuclear configuration

Introduce a set of orbital "velocities" $\dot{\psi}_1, \ldots, \dot{\psi}_{N_s}$, a time scale parameter μ, and a fictitious electronic temperature, T_e such that $T_e \ll T$.

Car-Parrinello Lagrangian:

$$\begin{aligned} L &= \mu \sum_{i=1}^{N_e} \langle \dot{\psi}_i | \dot{\psi}_i \rangle + \sum_{I=1}^{N} \frac{1}{2} M_I \dot{\mathbf{R}}_I^2 - E[\{\psi\}, \{\mathbf{R}\}] \\ &+ \sum_{i,j} \Lambda_{ij} \left(\langle \psi_i | \psi_j \rangle - \delta_{ij} \right) \end{aligned}$$

Car-Parrinello equations of motion:

$$\mu |\ddot{\psi}_i\rangle = -H_{\rm KS} |\psi_i\rangle + \sum_j \Lambda_{ij} |\psi_j\rangle$$

$$M_I \ddot{\mathbf{R}}_I = -\frac{\partial E}{\partial \mathbf{R}_I}$$

$$H_{\rm KS} = -\frac{1}{2}\nabla^2 + \int d\mathbf{r}' \, \frac{n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + \frac{\delta E_{\rm xc}}{\delta n(\mathbf{r})} + V_{\rm ext}(\mathbf{r}, \{\mathbf{R}\})$$

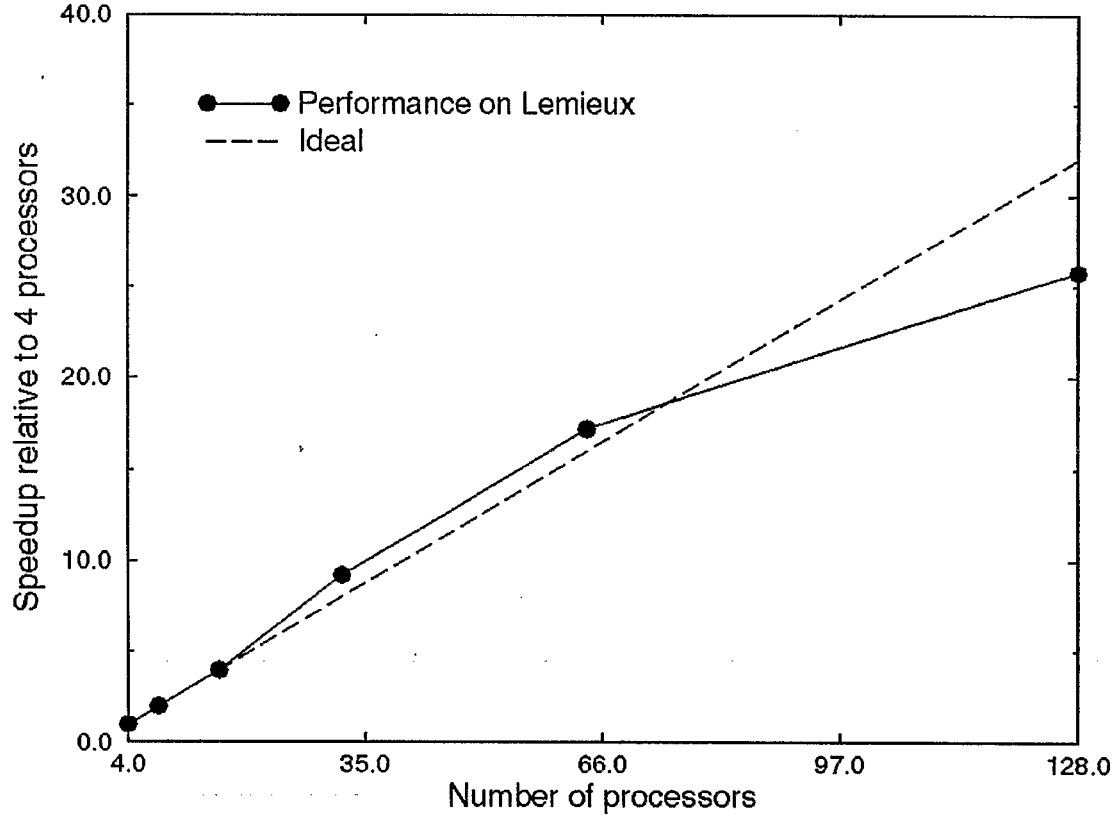
$$= -\frac{1}{2}\nabla^2 + V_{\rm H}(\mathbf{r}) + V_{\rm xc}(\mathbf{r}) + V_{\rm loc}(\mathbf{r}, \{\mathbf{R}\}) + \hat{V}_{\rm NL}(\{\mathbf{R}\})$$

Performance of PINY_MD code

MET, *et al*, *Comp. Phys. Comm.* **128**, 333 (2000)

SYSTEM:

64 waters in 12.43 Å box, DFT = BLYP, 256 KS states, 24,000 PWs per state



Ab initio molecular dynamics beyond plane waves

- Ab initio molecular dynamics employ plane wave basis:
  - **Expansion**: $\psi_i(x) = \frac{1}{\sqrt{L}} \sum_p c_p e^{ipx/\hbar}$
  - **Momentum eigenfunctions**:

$$P|p\rangle = p|p\rangle$$

$$\langle x|p\rangle = \frac{1}{\sqrt{2\pi\hbar}} e^{ipx/\hbar}$$

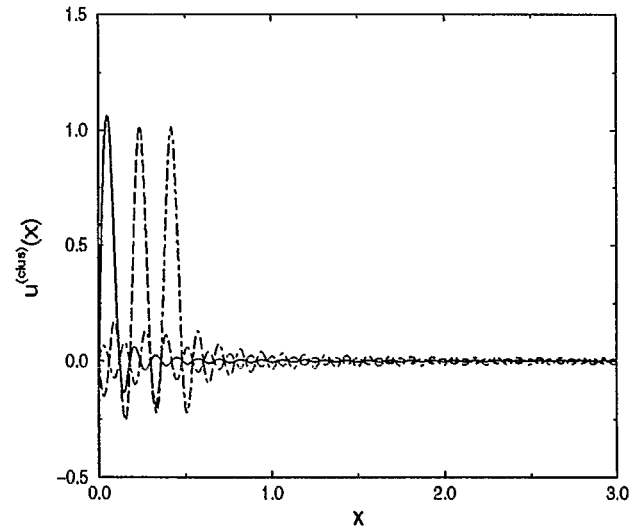
  - **FFT grid**: $\{x_1, ..., x_N\}$
  - **Good for long range interactions**.

- Discrete Variable Representations (DVRs):

J. C. Light, I. P. Hamilton and J. V. Lill, *J. Chem. Phys.* **82**, 1400 (1985).

D. T. Colbert and W. H. Miller, *J. Chem. Phys.* **96**, 1982 (1985).

  - **DVR grid**: $\{x_1, ..., x_N\}$
  - **Continuous functions**, $\{u_i(x)\}$, $i = 1, ..., N$
  - **Coordinate eigenfunctions**: $u_k(x_l) = \delta_{kl}/a_k$
  - **Expansion**: $\psi_i(x) = \frac{1}{\sqrt{L}} \sum_k C_k u_k(x)$
  - **Good for short range interactions**



- Optimal solution: **Combination**.

Results

Y. Liu and M. E. Tuckerman, *Phys. Rev. Lett.* (submitted)

System: 8 Si atoms in a 10 Å periodic box, LDA, BHS pseudopotentials.

Grid Size	E(PW)[1]	Δ_{PW}[2]	E(DVR)[1]	Δ_{DVR}[2]
16^3	-30.8337	610	-31.8094	2.4
20^3	-31.2026	378	-31.8057	0.06
32^3	-31.7936	8	-31.8056	0
48^3	-31.8050	0.4	-31.8056	0
60^3	-31.8051	0.3	-31.8056	0

[1] Hartrees

[2] kcal/mol

Grid Size	F(PW)[1]	Δ_{PW}[1]	F(DVR)[1]	Δ_{DVR}
16^3	101.49	97.54	13.32	13.32
20^3	12.83	8.88	3.92	0.03
32^3	3.89	0.06	3.97	0.02
48^3	3.97	0.02	3.95	0
60^3	3.96	0.01	3.95	0

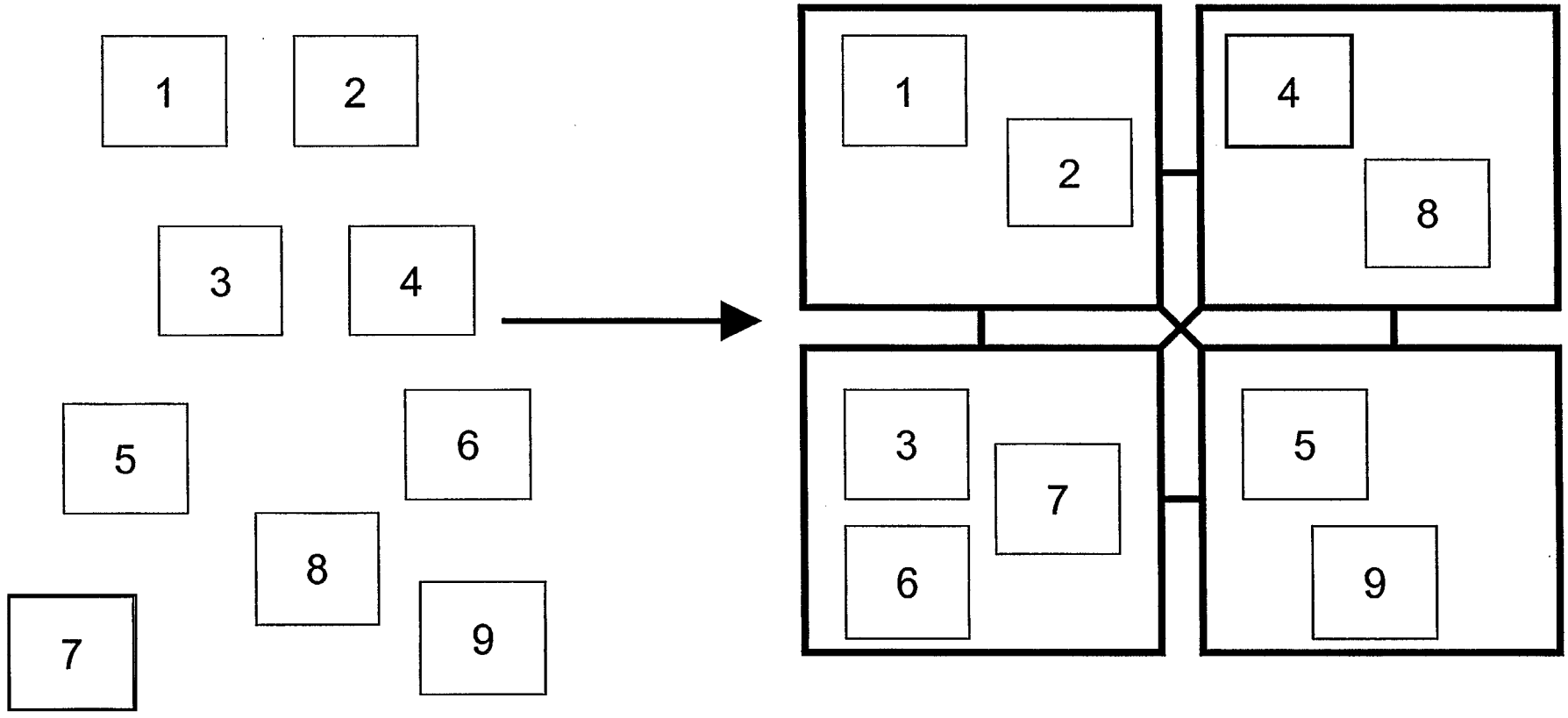
[1] kcal/(mol·Å)

System: 1 H_2O molecule in a 5 Å cluster box, BLYP, TM pseudopotentials.

Grid size	E(PW)[1]	Δ_{PW}[2]	E(DVR)[1]	Δ_{DVR}[2]
24^3	-15.1056	1311	-17.1222	45.6
32^3	-16.0058	746	-17.1769	11
40^3	-16.8569	212	-17.1939	0.6
48^3	-17.0183	110	-17.1948	0.06
60^3	-17.1776	11	-17.1949	0
80^3	-17.1938	0.6	-17.1949	0

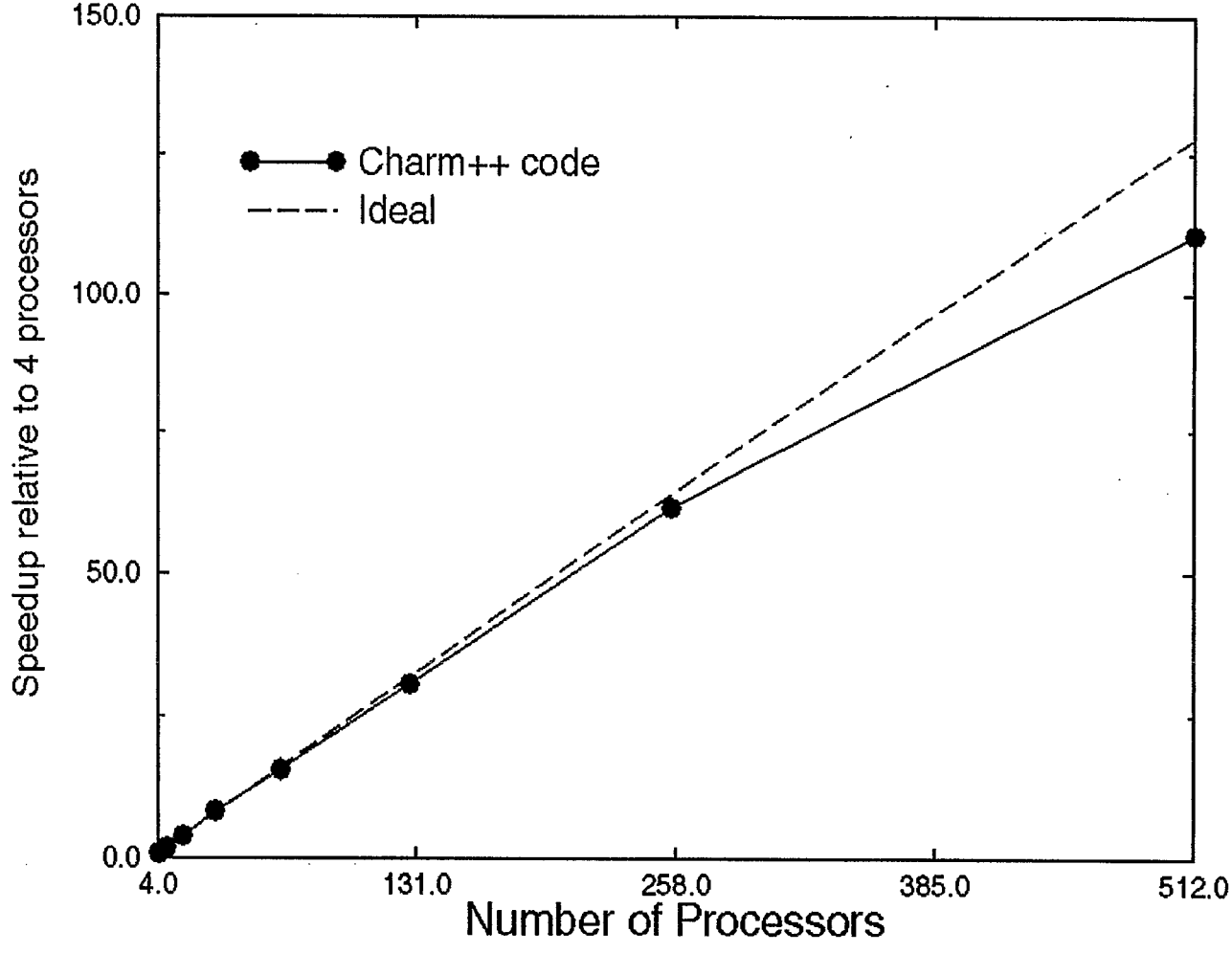
Computing with Charm++

***NSF sponsored collaboration with L. V. Kale at UIUC.*



N tasks or objects (chares) are dynamically mapped onto *P* processors at runtime in order to achieve optimal load balancing. Objects may be migrated during a run as processors as computing damands/resource availability change.

Preliminary results on Lemieux for 64-water system



BlueGene Future Directions

Alan Gara
IBM Research

There exist many similarities between QCDOC and BG/L. These similarities are based on many hardware features that are shared by both machines. These include the ability to do fast global operations in hardware as well as the common use of a torus for the network topology. These hardware features result in similar software development requirements. This spans system software as well as application software. On the application front, both machines have similar behavior with respect to message locality and in fact a similar process can be developed by which applications can be investigated as to how "appropriate" the application is for these architectures.

When one looks at application porting exercises that have been undertaken on BG/L to date, we find that the locality along with the application message size are the limiting factors for scalability. For problems that have long range communication patterns the message size will usually determine the scalability. For local problems such as nearest neighbor communication algorithms, the message sizes scale better but this will usually determine the limits to the application scalability. We have not found many applications that had scaling problems due to the bandwidth limitations. The general difficulty is that message overheads in terms of finite packet size as well as software overhead dominate the communication time. When this happens the performance drops dramatically with increasing numbers of processors. This point of where the "knee" of the performance scaling curve resides is application dependent but in general for long range communication patterns it can be extended by about a factor of 2 with a low latency messaging interface.

Commonality between BG/L and QCDOC

- Backbone of network is torus
  - Effective bandwidth is strong function of message locality
    - Potentially large performance gain for efficient mapping problem to machine.
    - Example: Adaptive methods would use similar optimization techniques
    - Parallel libraries , Frameworks ...
- Global operations supported in hardware
  - Applications can similarly leverage
- Fast Barrier/ Interrupt Capability
  - Applications can similarly leverage
- Same Integer Processing Core (440)
  - PowerPC Book E instruction set
  - MMU, TLBs ... Identical
    - Libraries
    - Computational kernel optimization

Commonality between BG/L and QCDOC (2)

- Lightweight OS kernel
  - Could leverage many kernel functions
    - memory management, host client functions, debugging support
- Stateless JTAG boot
  - Allows for simple reproducible application support model
    - Requirements on environment external to hardware very similar
- Likely to have similar failure modes
  - Much to learn from each others experience
    - soft error susceptibility
    - predictive failure analysis
    - machine diagnostics and operating "philosophy" leading to reliable, reproducible application performance
- Application viability
  - Both systems require that the application scales well with effective torus bandwidth constraints
    - Many aspects of application studies are applicable across platforms

Application porting insights

- Scaling on BG/L is limited mainly by two effects
  - Effective bandwidth decreases as scale increases
  - Message size decreases as scale increases
- Memory requirements
  - working set - scales down with increasing machine size
    - Often quickly
  - tables, constants - often large and small scaling effect
    - Largest contributor to node memory requirements
- For fixed problem scaling, message size is the strongest factor
  - Much stronger scaling behavior
- Software overheads will determine the "knee" for scaleability for fixed scaling
  - Latency induced bandwidth limitation will often dominate
    - Severe constraint calls for heroic effort to minimize effect

Application porting insights

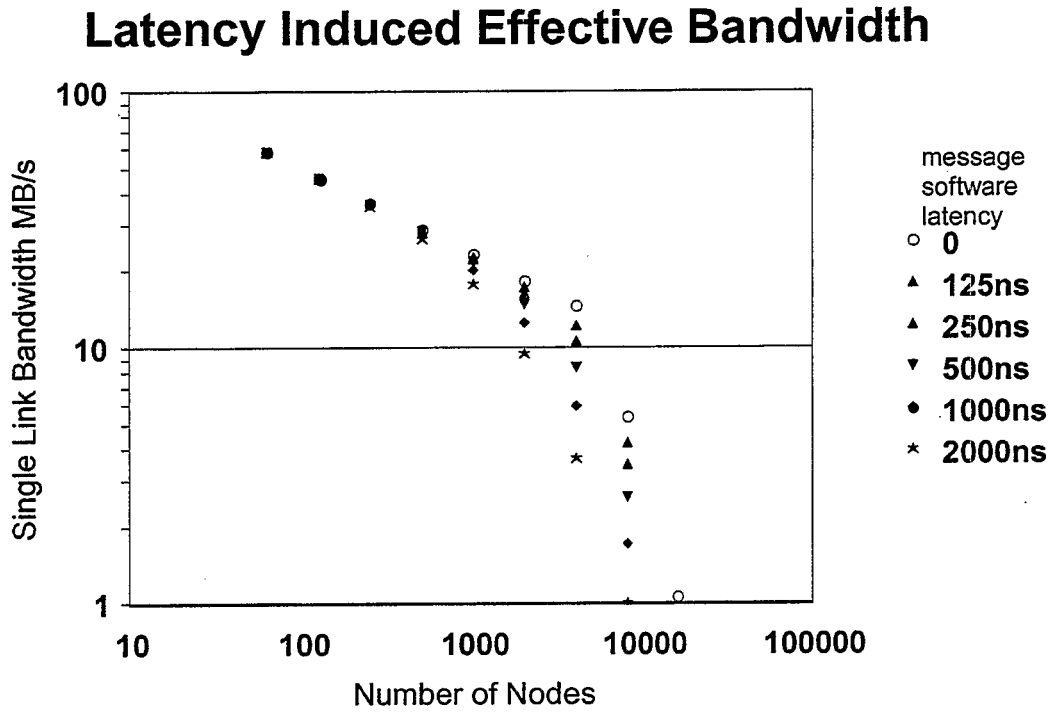
Scaling dependence for machine of size N^3=T

	3D nearest neighbor	all-to-all communication	2-D transpose (as required for 3d FFT)
Message size for fixed problem size scaling	N^(-2) T^(-2/3)	N^(-6) T^(-2)	N^(-5) T^(-5/3)
Message size for fixed size/node scaling	1	N^(-3) T^(-1)	N^(-2) T^(-2/3)
Effective bandwidth scaling	1	N^(-1) T^(-1/3)	N^(-1) T^(-1/3)

Hardware launch latency for minimum message size (32B)

Number of Nodes	3D nearest neighbor	all-to-all communication
64	30ns	90ns
512	30ns	180ns
4096	30ns	360ns
32k	30ns	720ns
64k	30ns	1440ns

All- to-all effective communication bandwidth (single link)
Message size is 1MB at 64 nodes; fixed problem size scaling



William D. Gropp
Mathematics and Computer Science Division
Argonne National Laboratory

Trends in High Performance Computing

This talk reviewed some of the recent trends in high performance computing, covering the interlinked aspects of hardware, software, and algorithms. The first part covered some of the challenges for computer hardware, including an increasing gap between CPU and memory speeds, CPU clock rates so fast that a clock signal can no longer cross a CPU chip in a single cycle, and power dissipation levels approaching that of a nuclear reactor. All of these hardware challenges suggest new directions, such as the massive parallelism in the QCDOC and BlueGene machines. The second part covered trends in software for high performance computing, including the increasing use of components, and recent successes in the partial automation of software generation and tuning. Examples drawn from the ATLAS tool show both the potential of these new tools and the limitations of a pure compiler-oriented approach. Because applications often take years to write, and most parallel HPC applications are already written in the message passing interface (MPI), features of MPI that are particularly appropriate for highly parallel and low-latency machines was also covered. The third part briefly discussed some of the features of the best algorithms, including adaptive and hierarchical algorithms, as well as opportunities for higher order methods and for algorithms with better numerical properties (e.g., in which 32-bit floating point or even fixed point arithmetic is sufficient). The talk concluded with some recommendations for addressing the open questions in the programming and use of massively parallel, processor rich systems such as QCDOC and BlueGene.

Trends in High Performance Computing

William D. Gropp
Mathematics and Computer Science
www.mcs.anl.gov/~gropp

Trends in HPC Software I

- Applications
  - Increasing use of components
    - Complexity of algorithms and multidisciplinary science makes "single Fortran file" programs unworkable
  - Lifetime of applications is either very short (a few days for some) or very long (many generations of computer hardware)
  - Lifetime-cost of applications can dominate

Trends in HPC Software II

- High Performance Languages, Libraries, and Middleware
  - Abandon the sequential consistency/PRAM model
    - Hardware *cannot* support that model efficiently
    - No evidence that systems can efficiently emulate this execution model
  - UPC, CoArray Fortran, MPI (both MPI-1 and MPI-2 Remote Memory Access (RMA))
    - All relax memory consistency, in a precise way
    - Even 100ns is a *long* time
  - Intrinsically Scalable Design
    - However, few implementations optimized for 16K+ (e.g., MPI buffer management)

MPI Comments

- MPI (the specification) designed for performance and scalability, e.g.,
  - "Zero copy" transfers
  - Weak message ordering (good support for unordered networks)
  - No* nonscalable data structures
    - E.g., no requirement for data buffers on each process for all other processes
  - Collective operations can exploit best hardware (need not be layered on MPI point-to-point)
  - MPI-I/O collective semantics critical for performance
  - Nonblocking operations hide latency (but not overhead)
- MPI-2 adds remote-memory access
  - Eliminates many sources of overhead, including message tag matching, buffer management; consistency model allows latency hiding
- MPI designed to enable libraries
  - Replacing MPI point-to-point with MPI RMA within libraries allows applications to benefit without rewriting
- MPI process topology routines allow an application to request and adapt to physical topology
  - Best fit to n-dimensional meshes, either periodic or not

Trends in HPC Software III

- *Partial* automation of software development, maintenance, and tuning, e.g.
  - ATLAS
  - Automatic Differentiation
  - Telescoping languages
  - "Compiled libraries"
- "Indirection" as the solution to all problems

QCDoC and BG/L

- Leverage existing software and environments
- Use tools to automatically transform applications
  - Source-to-source translation of MPI applications to remove library overhead
  - S2S tools to prune libraries of unused methods
  - ATLAS-style tools to generate better code for important computational kernels
  - Use tools for important *families* of operations, trading effectiveness for generality

High Performance Computing with QCDOC and BlueGene
BNL / Columbia / IBM / RIKEN BNL Research Center
February 28, 2003

Organizers: Norman Christ, Columbia; James Davenport, BNL; Yuefan Deng, Stony Brook/BNL, Alan Gara, IBM; James Glimm, BNL/Stony Brook; Robert Mawhinney, Columbia; Edward McFadden, BNL; Arnold Peskin; BNL; William Pulleyblank, IBM

PARTICIPANTS

Name	Mailing Address	E-mail Address
Carl Anderson	Biology – 463 Brookhaven National Laboratory Upton, NY 11973	cwa@bnl.gov
Anthony Baltz	Physics – 510A Brookhaven National Laboratory Upton, NY 11973	baltz@bnl.gov
Robert Bennett	ITD - 515 Brookhaven National Laboratory Upton, NY 11973	robertb@bnl.gov
Federico Berruto	Physics – 510A Brookhaven National Laboratory Upton, NY 11973	fberruto@bnl.gov
Eric Blum	ITD – 515 Brookhaven National Laboratory Upton, NY 11973	blum@bnl.gov
Peter Bond	ITD – 515 Brookhaven National Laboratory Upton, NY 11973	bond@bnl.gov
Kenneth Bowler	University of Edinburgh James Clerk Maxwell Bldg. The King's Buildings Edinburgh EH9 3JZ United Kingdom	kcb@ph.ed.ac.uk
Peter Boyle	Department of Physics Columbia University New York, NY 10027	pab@phys.columbia.edu

Michael Brown	University of Edinburgh EPCC King's Buildings Edinburgh EH9 3JZ United Kingdom	m.w.brown@ed.ac.uk
Xiaodan Cai	Applied Mathematics & Statistics SUNY, Stony Brook Stony Brook, NY 11794	xiaodan@ustc.edu.cn
Tameka Carter	ITD – Bldg. 515 Brookhaven National Laboratory Upton, NY 11973	ttcarter@bnl.gov
Praveen Chaudhari	IBM T.J. Watson Research Center 37-204, 1101 Kitchawan Road/Rt. 134 Yorktown Heights, NY 10598	chaudha@us.ibm.com
Dong Chen	IBM T.J. Watson Research Center 37-204, 1101 Kitchawan Road/Rt. 134 Yorktown Heights, NY 10598	chendong@us.ibm.com
George Liang-Tai Chiu	IBM P.O. Box 218 Yorktown Heights, NY 10598	gchiu@us.ibm.com
Norman Christ	Department of Physics Columbia University 2960 Broadway New York, NY 10027-6902	nhc@phys.columbia.edu
Paul Cotues	IBM T.J. Watson Research Center 37-204, 1101 Kitchawan Road/Rt. 134 Yorktown Heights, NY 10598	cotues@us.ibm.com
Michael Cruetz	Physics – 510A Brookhaven National Laboratory Upton, NY 11973	creutz@bnl.gov
James Davenport	CDIC – 463B Brookhaven National Laboratory Upton, NY 11973	daven@bnl.gov
Yuefan Deng	Applied Mathematics & Statistics SUNY Stony Brook Stony Brook, NY 11794	deng@ams.sunysb.edu
Zihua Dong	Department of Physics Columbia University 2960 Broadway New York, NY 10027-6902	dong@phys.columbia.edu

Bruce Elmegreen	IBM T.J. Watson Research Center 37-204, 1101 Kitchawan Road/Rt. 134 Yorktown Heights, NY 10598	bje@us.ibm.com
Stratos Efstathiadis	ITD – Bldg. 515 Brookhaven National Laboratory Upton, NY 11973	stratos@bnl.gov
Irwin Gaines	CD-Computing Division Office Fermilab P.O. Box 500 Batavia, IL 60510-0500	gaines@fnal.gov
Alan Gara	IBM T.J. Watson Research Center 37-204, 1101 Kitchawan Road/Rt. 134 Yorktown Heights, NY 10598	alangara@us.ibm.com
Robert Germain	IBM T.J. Watson Research Center 37-204, 1101 Kitchawan Road/Rt. 134 Yorktown Heights, NY 10598	rgermain@us.ibm.com
James Glimm	CDIC – 463B Brookhaven National Laboratory Upton, NY 11973-5000	glimm@ams.sunysb.edu
William Gropp	Argonne National Laboratory 9700 S. Cass Avenue Argonne, IL 60439	gropp@mcs.anl.gov
Manish Gupta	IBM T.J. Watson Research Center 37-204, 1101 Kitchawan Road/Rt. 134 Yorktown Heights, NY 10598	mgupta@us.ibm.com
Donald J. Johann, Jr.	NCI-FDA Chemical Proteomics Project National Institutes of Health 8800 Rockville Pike Building 29A, Room 2A-21 Bethesda, MD 20892	johann@cber.fda.gov
Chulwoo Jung	Physics – 510A Brookhaven National Laboratory Upton, NY 11973	chulwoo@phys.columbia.edu
Richard Kenway	University of Edinburgh King's Buildings Edinburgh EH9 3JZ United Kingdom	r.d.kenway@ed.ac.uk
David Keyes	Department of Physics Columbia University New York, NY 10027	keyes@cs.odu.edu

Thomas Kirk	Physics – 510 F Brookhaven National Laboratory Upton, NY 11973	tkirk@bnl.gov
Lynn Kissel	Lawrence Livermore National Lab. L-060 P.O. Box 808 Livermore, CA 94551	lkissel@llnl.gov
T.D. Lee	Physics – 510A Brookhaven National Laboratory Upton, NY 11973	tdl@cuphyb.phys.columbia.edu
Thomas Liebsch	IBM 37th Street NW & Hwy. 52 Rochester, MN 55901	liebsch@us.ibm.com
Huey-Wen Lin	Department of Physics Columbia University New York, NY 10027	hwlin@phys.columbia.edu
Guofeng Liu	Department of Physics Columbia University New York, NY 10027	gl159@columbia.edu
Robert Mawhinney	Department of Physics Columbia University 538 W. 120th Street New York, NY 10027	rdm@physics.columbia.edu
Edward McFadden	ITD – 515 Brookhaven National Laboratory Upton, NY 11973	emc@bnl.gov
Michael McGuigan	ITD – 515 Brookhaven National Laboratory Upton, NY 11973	mcguigan@bnl.gov
Jose Moreira	IBM T.J. Watson Research Center 37-204, 1101 Kitchawan Road/Rt. 134 Yorktown Heights, NY 10598	jmoreira@us.ibm.com
Eric Myra	Dept. of Physics & Astronomy SUNY Stony Brook Stony Brook, NY 11794-3800	emyra@mail.astro.sunysb.edu
Junichi Noaki	Physics – 510A Brookhaven National Laboratory Upton, NY 11973	noaki@rccp.tsukuba.ac.jp

Shigemi Ohta	Physics – 510A Brookhaven National Laboratory Upton, NY 11973	ohta@bnl.gov
Konstantinos Orginos	Physics – 510A Brookhaven National Laboratory Upton, NY 11973	orginos@bnl.gov
Satoshi Ozaki	RHIC – 1005S Brookhaven National Laboratory Upton, NY 11973	ozaki@bnl.gov
Peter Paul	Director's Office – 460 Brookhaven National Laboratory Upton, NY 11973	ppaul@bnl.gov
Konstantin Petrov	Physics – 510A Brookhaven National Laboratory Upton, NY 11973	petrov@bnl.gov
Arnold Peskin	ITD – 515 Brookhaven National Laboratory Upton, NY 11973	peskin@bnl.gov
William Pulleyblank	IBM Research Division ESS P.O. Box 218 Yorktown Heights, NY 10598	wp@us.ibm.com
Claudio Rebbi	Physics Department Boston University 590 Commonwealth Avenue Boston, MA 02215	rebbi@bu.edu
Peter Rissland	Applied Mathematics & Statistics SUNY Stony Brook Stony Brook, NY 11794	rissland@ams.sunysb.edu
Nicholas Samios	Physics – 510A Brookhaven National Laboratory Upton, NY 11973	samios1@bnl.gov
David Stampf	ITD – 515 Brookhaven National Laboratory Upton, NY 11973	drs@bnl.gov
Takanori Sugihara	Physics – 510A Brookhaven National Laboratory Upton, NY 11973	sugihara@bnl.gov

Douglas Swesty	Dept. of Physics & Astronomy SUNY Stony Brook Stony Brook, NY 11794-3800	dswesty@mail.astro.sunysb.edu
Hiroshi Takahashi	Energy Sci. & Technology – 475B Brookhaven National Laboratory Upton, NY 11973	takahash@bnl.gov
Stanimire Tomov	ITD – 515 Brookhaven National Laboratory Upton, NY 11973	tomov@bnl.gov
Mark Tuckerman	Dept. of Chemistry New York University 100 Washington Square East New York, NY 10003	mark.tuckerman@nyu.edu
Tilo Wettig	Dept. of Physics Yale University New Haven, CT 06520	tilo.wettig@yale.edu

High Performance Computing with QCDOC and BlueGene

Brookhaven National Laboratory
Bldg. 490 Seminar Room
February 28, 2003

Workshop Agenda

Introduction (Chair, Arnie Peskin)

8:30 Welcome by T. D. Lee - RBRC, Bill Pulleyblank - IBM, and Tom Kirk - BNL

Session 1: System Development Status (Chair, Robert Mawhinney)

9:00 QCDOC System Overview and Status – Norman Christ Columbia
9:30 QCDOC Software Developments – Peter Boyle, Columbia & Dave Stampf, Rob Bennett, BNL

10:15 coffee break

10:30 BlueGene/L Hardware Overview – Dong Chen, IBM
11:00 BlueGene/L Software Overview – Jose Moreira, IBM

Session 2: Prospective Applications (Chair, Yuefan Deng)

11:30 Microsecond MD Simulation on QCDOC – Jim Glimm BNL/Stony Brook
12:00 Applications of QCDOC to Astrophysics – Doug Swesty, Stony Brook

12:30 – 1:30 Lunch Break

1:30 Life Sceinces and Other Applications on Blue/Gene - Bob Germain, IBM
2:00 Performance Stress-points for Parallel Implicit PDEs – David Keyes, Old Dominion University
2:30 Optimal schemes for Car-Parrinello based ab initio molecular dynamics on parallel architectures. – Mark Tuckerman, NYU

3:00 coffee break

Session 3: Future Plans (Chair, Jim Davenport)

3:15 BlueGene Future Directions – Al Gara
3:45 Trends in High Performance Computing Architecture – Bill Gropp
4:15 Prospects for Future Collaboration, Wrap-up – discussion
5:00 Adjourn

Additional RIKEN BNL Research Center Proceedings:

Volume 49 – RBRC Scientific Review Committee Meeting – BNL-52679
Volume 48 – RHIC Spin Collaboration Meeting XIV – BNL-
Volume 47 – RHIC Spin Collaboration Meetings XII, XIII – BNL-71118-2003
Volume 46 – Large-Scale Computations in Nuclear Physics using the QCDOC – BNL-52678
Volume 45 – Summer Program: Current and Future Directions at RHIC – BNL-71035
Volume 44 – RHIC Spin Collaboration Meetings VIII, IX, X, XI – BNL-71117-2003
Volume 43 – RIKEN Winter School – Quark-Gluon Structure of the Nucleon and QCD – BNL-52672
Volume 42 – Baryon Dynamics at RHIC – BNL-52669
Volume 41 – Hadron Structure from Lattice QCD – BNL-52674
Volume 40 – Theory Studies for RHIC-Spin – BNL-52662
Volume 39 – RHIC Spin Collaboration Meeting VII – BNL-52659
Volume 38 – RBRC Scientific Review Committee Meeting – BNL-52649
Volume 37 – RHIC Spin Collaboration Meeting VI (Part 2) – BNL-52660
Volume 36 – RHIC Spin Collaboration Meeting VI – BNL-52642
Volume 35 – RIKEN Winter School – Quarks, Hadrons and Nuclei – QCD Hard Processes and the Nucleon Spin – BNL-52643
Volume 34 – High Energy QCD: Beyond the Pomeron – BNL-52641
Volume 33 – Spin Physics at RHIC in Year-1 and Beyond – BNL-52635
Volume 32 – RHIC Spin Physics V – BNL-52628
Volume 31 – RHIC Spin Physics III & IV Polarized Partons at High Q^2 Region – BNL-52617
Volume 30 – RBRC Scientific Review Committee Meeting – BNL-52603
Volume 29 – Future Transversity Measurements – BNL-52612
Volume 28 – Equilibrium & Non-Equilibrium Aspects of Hot, Dense QCD – BNL-52613
Volume 27 – Predictions and Uncertainties for RHIC Spin Physics & Event Generator for RHIC Spin Physics III – Towards Precision Spin Physics at RHIC – BNL-52596
Volume 26 – Circum-Pan-Pacific RIKEN Symposium on High Energy Spin Physics – BNL-52588
Volume 25 – RHIC Spin – BNL-52581
Volume 24 – Physics Society of Japan Biannual Meeting Symposium on QCD Physics at RIKEN BNL Research Center – BNL-52578
Volume 23 – Coulomb and Pion-Asymmetry Polarimetry and Hadronic Spin Dependence at RHIC Energies – BNL-52589
Volume 22 – OSCAR II: Predictions for RHIC – BNL-52591
Volume 21 – RBRC Scientific Review Committee Meeting – BNL-52568
Volume 20 – Gauge-Invariant Variables in Gauge Theories – BNL-52590
Volume 19 – Numerical Algorithms at Non-Zero Chemical Potential – BNL-52573
Volume 18 – Event Generator for RHIC Spin Physics – BNL-52571

Additional RIKEN BNL Research Center Proceedings:

Volume 17 – Hard Parton Physics in High-Energy Nuclear Collisions – BNL-52574
Volume 16 – RIKEN Winter School - Structure of Hadrons - Introduction to QCD Hard Processes – BNL-52569
Volume 15 – QCD Phase Transitions – BNL-52561
Volume 14 – Quantum Fields In and Out of Equilibrium – BNL-52560
Volume 13 – Physics of the 1 Teraflop RIKEN-BNL-Columbia QCD Project First Anniversary Celebration – BNL-66299
Volume 12 – Quarkonium Production in Relativistic Nuclear Collisions – BNL-52559
Volume 11 – Event Generator for RHIC Spin Physics – BNL-66116
Volume 10 – Physics of Polarimetry at RHIC – BNL-65926
Volume 9 – High Density Matter in AGS, SPS and RHIC Collisions – BNL-65762
Volume 8 – Fermion Frontiers in Vector Lattice Gauge Theories – BNL-65634
Volume 7 – RHIC Spin Physics – BNL-65615
Volume 6 – Quarks and Gluons in the Nucleon – BNL-65234
Volume 5 – Color Superconductivity, Instantons and Parity (Non?)-Conservation at High Baryon Density – BNL-65105
Volume 4 – Inauguration Ceremony, September 22 and Non -Equilibrium Many Body Dynamics – BNL-64912
Volume 3 – Hadron Spin-Flip at RHIC Energies – BNL-64724
Volume 2 – Perturbative QCD as a Probe of Hadron Structure – BNL-64723
Volume 1 – Open Standards for Cascade Models for RHIC – BNL-64722

For information please contact:

Ms. Pamela Esposito
RIKEN BNL Research Center
Building 510A
Brookhaven National Laboratory
Upton, NY 11973-5000 USA

Phone: (631) 344-3097
Fax: (631) 344-4067
E-Mail: pesposit@bnl.gov

Ms. Tammy Heinz
RIKEN BNL Research Center
Building 510A
Brookhaven National Laboratory
Upton, NY 11973-5000 USA

(631) 344-5864
(631) 344-2562
theinz@bnl.gov

Homepage: http://www.bnl.gov/riken

RIKEN BNL RESEARCH CENTER

High Performance Computing with QCDOC and BlueGene

February 28, 2003

Li Keran

Nuclei as heavy as bulls
Through collision
Generate new states of matter.
T.D. Lee

Speakers:

R. Bennett
A. Gara
D. Keyes
D. Swesty
P. Boyle
R. Germain
T.D. Lee
M. Tuckerman
D. Chen
J. Glimm
J. Moreira
N. Christ
W. Gropp
D. Stampf

Organizers: N. Christ, Y. Deng, A. Gara, J. Glimm, R. Mawhinney, E. McFadden, A. Peskin, W. Pulleyblank